# PE Practical Electronics

The UK's premier electronics, computing and maker magazine

**Max's Cool Beans**
The classic Snake game & a VFD timer

**Teach-In 12**
Part 7: spectrum and vector network analysis

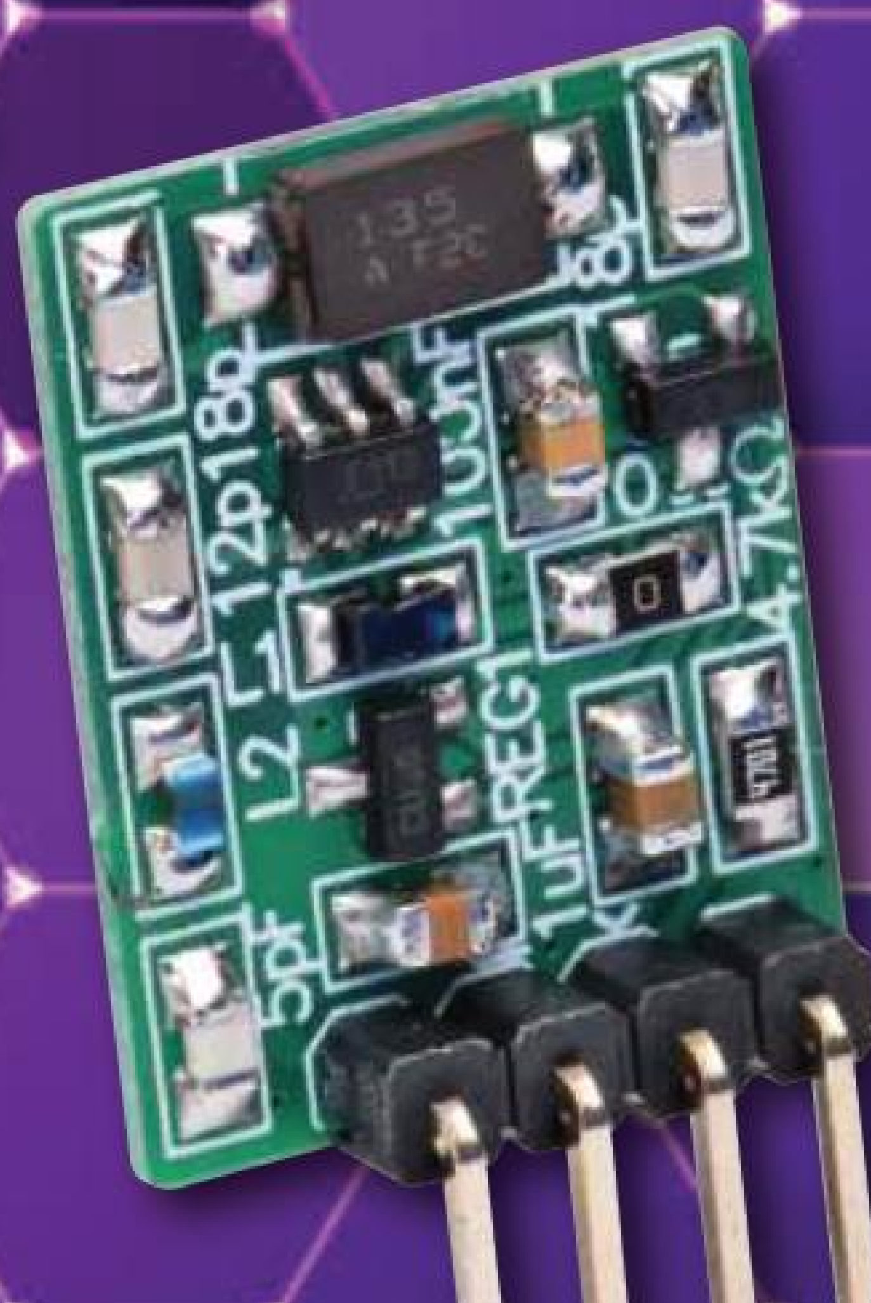
**Audio Out**
All-pass filters for speaker alignment

**Circuit Surgery**
Better ways to stabilise op amps

SSB RECEIVER
3.609.000 MHz
LSB
ANT TUNE
RF GAIN
BAND
SQUELCH
USB
ON
LSB
FAST
AGC
SLO
VOLUME
PHONES

## Build a Single Sideband Shortwave Receiver with digital tuning

## A motion-sensing Safety Timer for tools like soldering irons



## DIY LPD433 transmitter module

**PLUS!**

Power Electronics, part two – Understanding DC/DC converters

Techno Talk – Tensor processors and Raspberry Pi AI HATs

The Fox Report – Freely's success will depend on good hardware

May 2026 £5.99

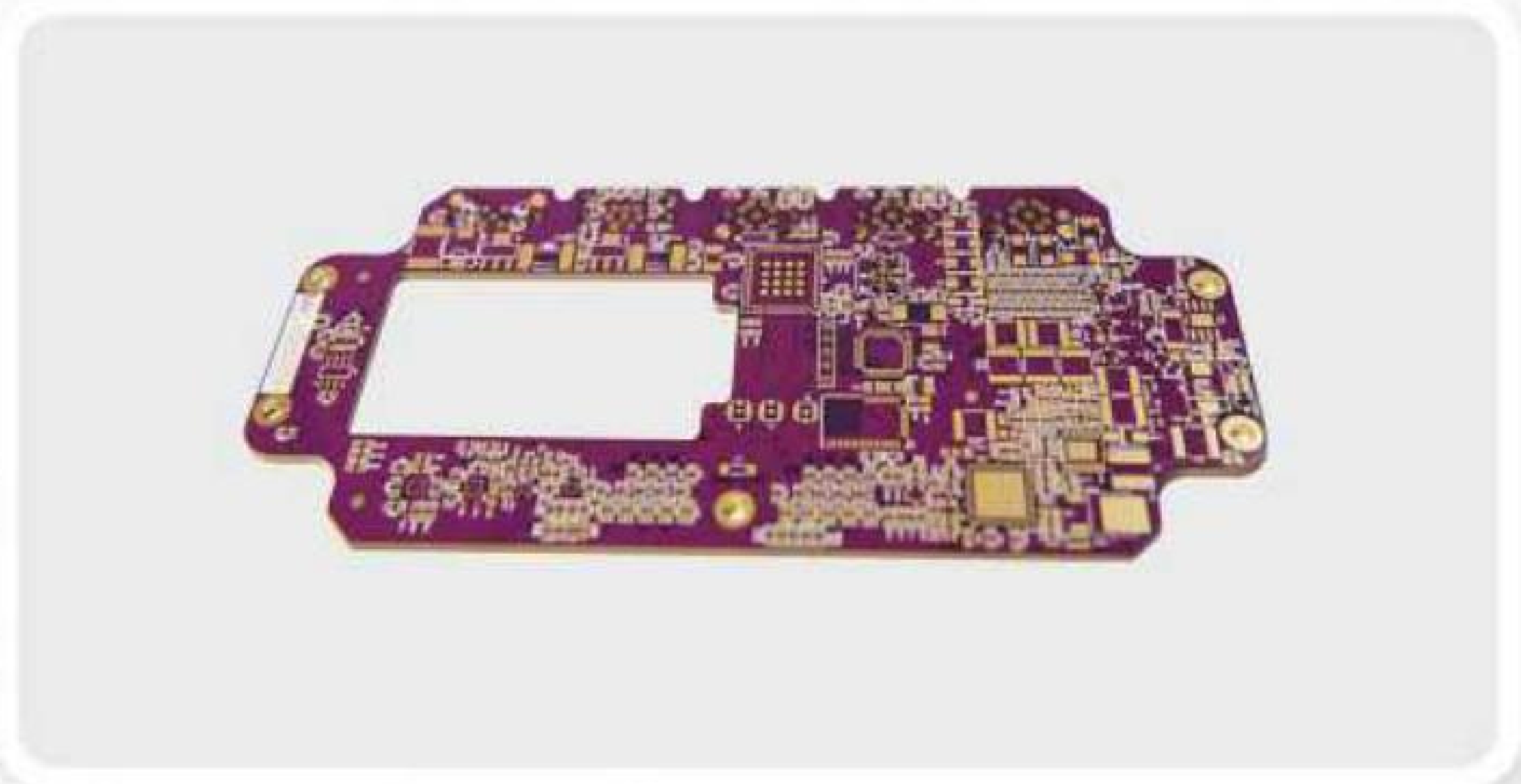
www.electronpublishing.com | @practicalelec | practicalelectronics

**PCBWay**

## Beyond Standard —
# Discover PCBWay's Advanced PCB Lineup

## About PCBWay:

With dedicated workshops for advanced PCBs, PCBWay offers full-spec, high-precision, and large-scale production. We ensure customers are supplied with PCBs that meet stringent quality standards at the lowest cost.

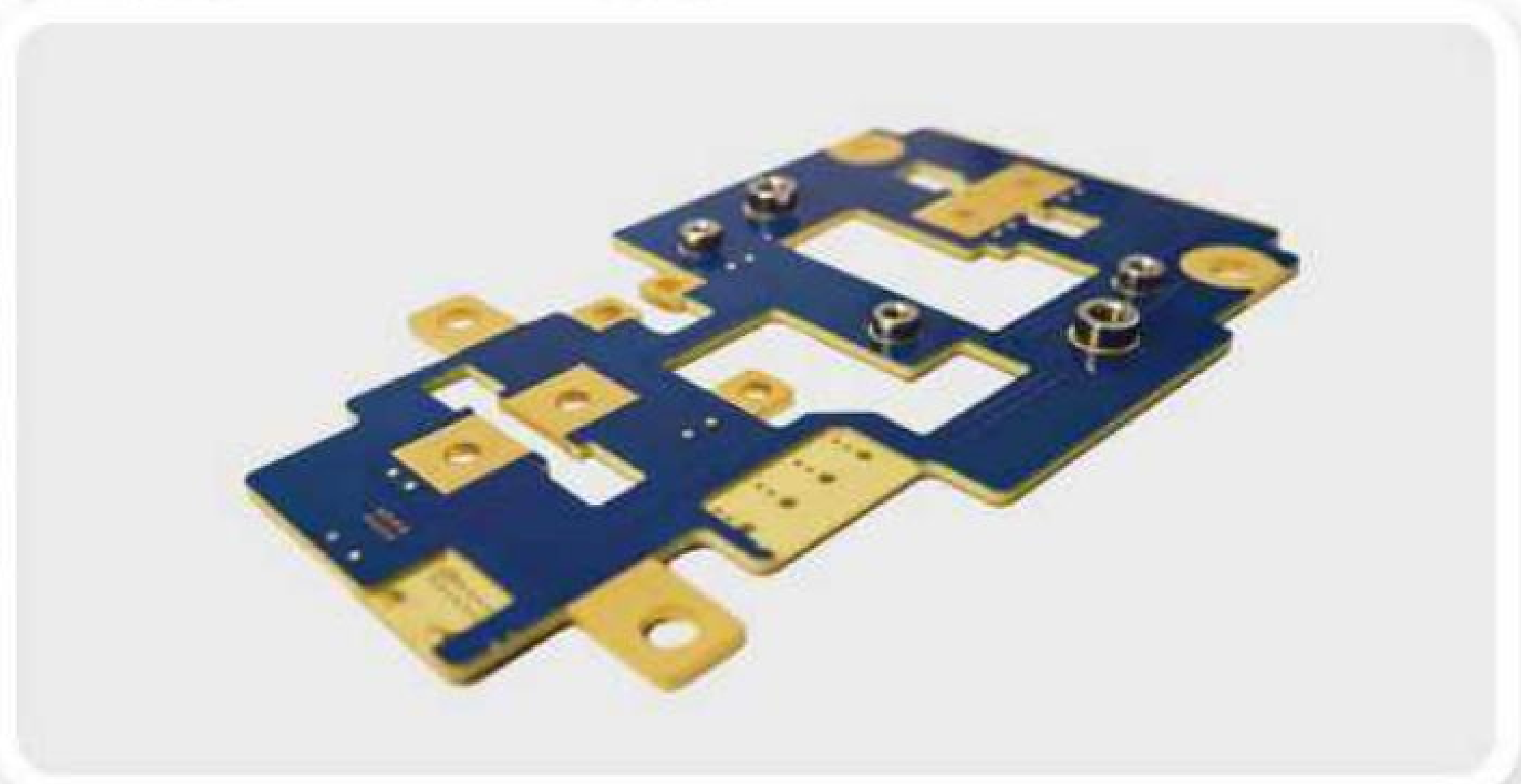


### HDI PCB //

- 4-60 layers
- FR-4, High Tg, Halogen-free, PTFE
- Finished Copper: 1/3OZ–8OZ
- Min Track/Spacing (inner): 2/2mil (H/HOZ base copper)
- Min BGA Pitch: 0.35mm

### High Frequency PCB //

- RO4003C, RO4350B, Ro3003, Ro3010, RT5880
- Min. Track/Spacing: 2mil
- Min. Hole Size: 0.15mm
- Finished Cooper: 1-2oz
- Board Thickness: 0.2–3.2mm



### Other PCBs with Special Process //

- Embedded Copper PCB
- Backplane PCB
- Cavity PCB
- White/Partially Hybrid PCB
- Black Core PCB (After-dark)

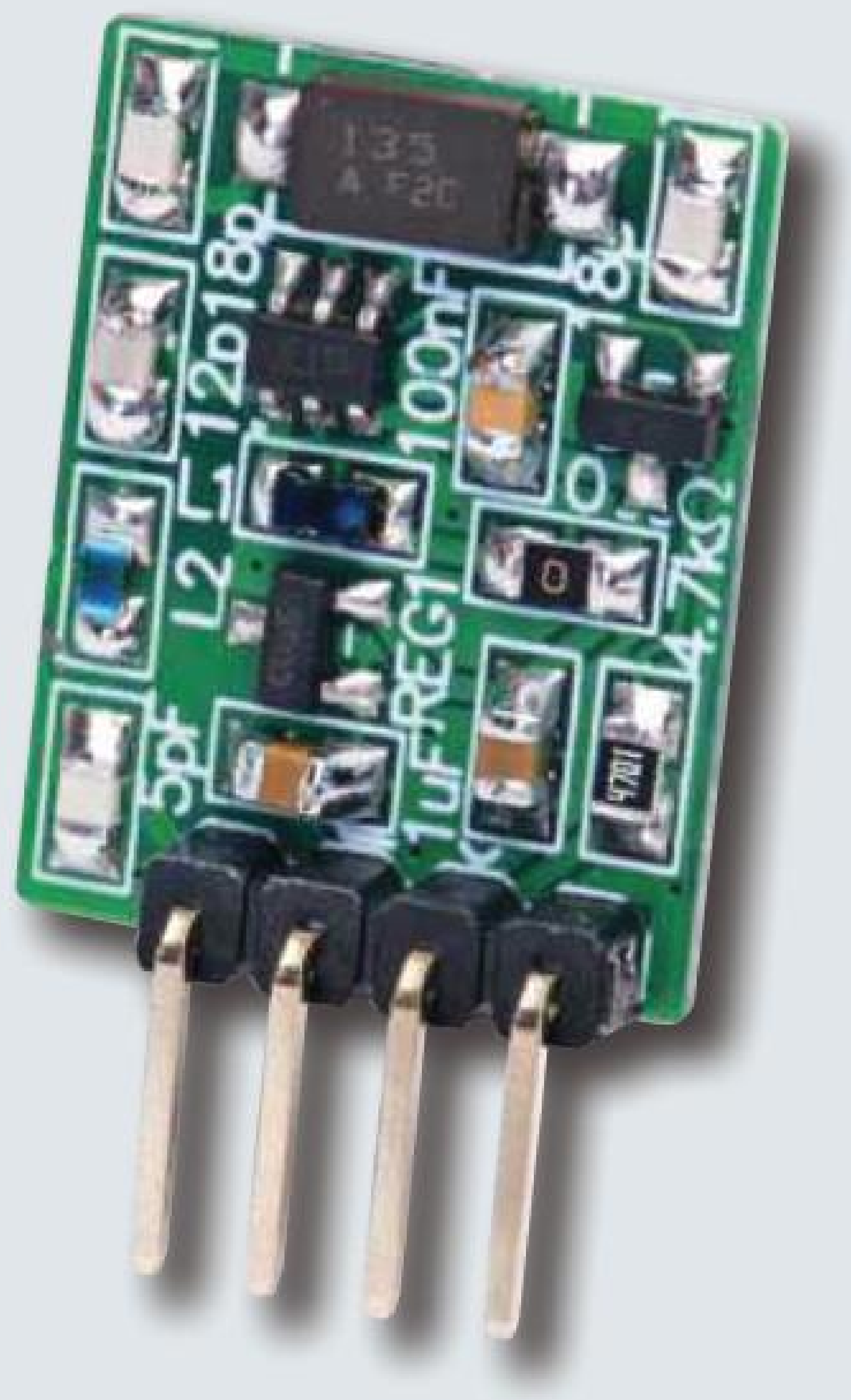
www.pcbway.com service@pcbway.com

# Practical Electronics

Volume 55. No. 05
May 2026
ISSN 2632 573X

# Contents

7.100.000 MHz
LSB
ANT TUNE
RF GAIN
BAND
SQUELCH
VOLUME
ON/OFF
USB/LSB
FAST/SLOW
CSE250202



## Projects and Circuits

**SSB Shortwave Receiver, part one** by Charles Kosina 26
While there are plenty of cheap radios these days, including software-defined types, I decided to build an analog shortwave radio myself. I learned a lot about shortwave, SSB and how radios work in the process.

**433MHz Transmitter Module** by Tim Blythman 46
The LPD433 (low-interference potential device) band is used by many devices for radio control. There is no need for a class licence, so low-cost 433-434MHz digital modules make it easy to create a wireless link.

**Tool Safety Timer** by Phil Prosser 60
Have you ever accidentally left your soldering iron switched on? This project will help allay your fears. It's also a great idea for fitting to a hot air rework system or other mains-powered tools.

## Series, Features and Columns

**Net Work** by Alan Winstanley 3
This month's column uncovers a common fault in a popular 'smart' charger and examines a USB Power Delivery (PD) adaptor for USB chargers and power banks. We also have a heads-up on looming import duty changes.

**Max's Cool Beans** by Max The Magnificent 6
Weird & Wonderful Arduino Projects, part 17: chronological contraptions & solid snakes. This month, we make further progress with our 'retro' games machine, making the classic Snake actually playable. I also discuss plans for a new VFD-based timer.

**The Fox Report** by Barry Fox 13, 25
Freely's 'live TV' streaming success will depend on good hardware support.

**Teach-In 12.7: the World of Wireless** by Mike Tooley 14
This instalment is on spectrum and vector network analysis (VNA). VNA enables the accurate measurement of radio-frequency (RF) network and device characteristics over a wide frequency range.

**Circuit Surgery** by Ian Bell 38
Measuring circuit frequency response using a PC sound card, part 8: more on op amp compensation. We look at better methods to stabilise op amp circuits that don't impact their DC accuracy and may give better phase margin.

**Audio Out** by Jake Rothman 54
All-pass filters are very useful for time alignment compensation in loudspeakers. This article includes some detailed examples of their usage in various products.

**Techno Talk** by Max the Magnificent 67, 75
From humble 8-bit microprocessors and Arduino boards to chiplets, XPUs, stacked silicon and edge-AI accelerators, a ramble through the astonishing (and slightly bewildering) evolution of modern computing.

**Power Electronics, part two** by Andrew Levido 68
This series continues by delving into the control systems required to make DC/DC converters work correctly.

## Regulars and Services

**Editorial** 2
Product codes should be specific to the hardware
**Reader services – Editorial and Advertising Departments** 2
**Electronics Teach-In Bundles** 45
**Practical Electronics PCB Service** 76-78
PCBs for Practical Electronics projects
**Classified ads and Advertiser index** 79
**Next month! – highlights of our next issue of *Practical Electronics*** 80

**Made in Australia & the UK.**

**Written in Britain, Australia, the USA and Ireland.**

**Read everywhere.**

© Silicon Chip Publications Pty Ltd 2026

Copyright in all drawings, photographs, articles, technical designs, software and intellectual property published in *Practical Electronics* is fully protected, and reproduction or imitation in whole or in part are expressly forbidden.

The June 2026 issue of *Practical Electronics* will be published on Tuesday, 5th of May, 2026 (see p80).

Renewals paid by PayPal, Visa and Mastercard are now handled via our website:
www.electronpublishing.com

# Practical Electronics

Volume 55. No. 05
May 2026
ISSN 2632 573X

# Editorial

**Editorial offices**
Electron Publishing (Australia)
**Tel** +61 2 9939 3295
**Email** pe@pemag.au
**Web** www.electronpublishing.com

**Address mail to:**
Electron Publishing (Australia)
PO Box 194, Matraville NSW 2036
Australia

**Advertising enquiries**
+61 2 9939 3295 pe@pemag.au

| | |
|---|---|
| **Editor** | Nicholas Vinen |
| **Publisher** | Nicholas Vinen |
| **Digital subscriptions** | Stewart Kearn **Tel** **07918 614662** |
| **Online Editor** | Alan Winstanley |
| **Web Systems** | Kris Thain |
| **Production** | Bao Smith |
| **Technical staff** | Tim Blythman, John Clarke |

**Print subscriptions**
Practical Electronics Subscriptions
PO Box 6337
Bournemouth BH1 9EH
United Kingdom
**Tel** 01202 087631
**Email** pesubs@selectps.com

**Technical enquiries**
We regret that technical enquiries cannot be answered over the telephone. We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine.

Questions about articles or projects should be sent to the editor by email: pe@pemag.au

**Projects and circuits**
All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it.

Some projects and circuits published in *Practical Electronics* employ voltages that can be lethal. Do not build, test, modify or fix any mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD (GFCI) adaptor.

**Component supplies**
Silicon Chip Publications may offer kits or other parts for making our projects, but not in all cases. When kits are not available, readers will need to find and source parts themselves. We advise readers to check that all parts are still available before commencing any project in a back-dated issue.

**Advertisements**
Although the proprietors and staff of *Practical Electronics* take reasonable precautions to protect the interests of readers by ensuring as far as practicable that advertisements are bona fide, the magazine and its publishers cannot give any undertakings in respect of statements or claims made by advertisers, whether these advertisements are printed as part of the magazine, or in inserts. The Publishers regret that under no circumstances will the magazine accept liability for non-receipt of goods ordered, or for late delivery, or for faults in manufacture.

**Transmitters/bugs/telephone equipment**
We advise readers that certain items of radio transmitting and telephone equipment which may be advertised in our pages cannot be legally used in the UK. Readers should check the law before buying any transmitting or telephone equipment, as a fine, confiscation of equipment and/or imprisonment can result from illegal use or ownership. The laws vary from country to country; readers should check local laws.

## Product codes should be specific to the hardware

One of the more frustrating aspects of computers and electronics is the way some manufacturers will ‘silently’ change the hardware of a product while keeping the same model name and number. This practice is misleading to consumers and is particularly common in the Wi-Fi and SSD (solid-state drive) markets.

For example, you might read a review of the Oceandoor FlameSturgeon 18000X SSD stating that it is exceptionally fast. On that basis, you buy one, install it, and discover that its performance falls well short of expectations. Why?

Unbeknownst to you, Oceandoor may have had difficulty sourcing the Feefifofum 300 controller used in the original review sample, and quietly substituted it with the inferior Feefifofum 200. The PCB was redesigned accordingly, and the resulting product – slower and cheaper to manufacture – was sold under the same model number.

Manufacturers usually ‘cover themselves’ by making vague claims such as “up to 5000MB/s”, so that when this sort of switcheroo occurs, the revised product still meets its advertised specifications. Nevertheless, it’s a poor practice and one that I believe misleads consumers.

In my view, a specific model number should correspond to a specific hardware configuration. Only very minor revisions (those that have negligible impact on performance, or that genuinely improve it) should be allowed under the same model number. Anything more substantial should require a new designation. In the example above, calling the revised product the FlameSturgeon 17000X, 18000XB or 18000X-2 would at least make it clear that it is not identical to the original. I don’t think that would stop people from buying it.

I’ve encountered similar problems with Wi-Fi dongles. Years ago, I would read reports that a particular dongle worked well under Linux, buy the same model, and find that it wasn’t recognised at all. The reason, once again, was a silent hardware revision that used a completely different and incompatible chipset.

Sadly, this still happens today. It shouldn’t. While it may not rise to the level of fraud or outright false advertising, it is certainly not consumer-friendly behaviour. If manufacturers are unwilling to address the problem voluntarily, it may ultimately require legislation to ensure that product codes actually mean something.

**Nicholas Vinen, Electron Publishing (Australia)***
* a division of Silicon Chip Publications Pty Ltd.

# Net Work

Alan Winstanley

This month's column uncovers a common fault in a popular 'smart' charger and examines a USB Power Delivery (PD) adaptor for USB chargers and power banks. We also have a heads-up on looming import duty changes.

**In the last *Net Work* column** (December 2025), I described how I was looking after my sealed lead-acid (SLA) 'Cyclon' type lawnmower starting battery, ready for the onset of winter. This winter I've used a Sealey Autocharge 100HF compact 'smart' charger to float-charge it, nurturing the battery in readiness for the spring mowing season.

In between times, my car battery had also been topped up using a popular CTEK MXS 5.0 12V 5A charger during periods of intermittent heavy use. These electronic chargers cost about £80; CTEK has a wide range of connecting leads and under-bonnet adaptors suitable for car and marine use.

On a previous car, I even fitted a CTEK connector to the car battery posts, under the clamp nuts, which made routine charging a cinch.

Despite my best endeavours, events took a turn for the worse when my car dashboard started flashing up "Stop-start not available", even though I knew the battery was fully charged. (I always disable stop-start anyway.) I even demonstrated this to the valet who collected the vehicle for service.

My dealer shrugged it off and reported that the battery tested OK, but bitter experience told me that the constant error message was a warning sign of imminent battery failure. I'd seen exactly the same thing three years earlier, when the car battery suddenly failed and I was left stranded in town late one evening.

It seems that an enhanced flooded battery (EFB) has a lifespan of just 3-5 years and, typically, mine was out of warranty by two weeks. *[Editor's note – we replaced the original battery in our Mazda CX-9 nearly seven years after taking delivery. It seems OEM battery quality is well above that of replacement batteries, even from the original dealer.]*

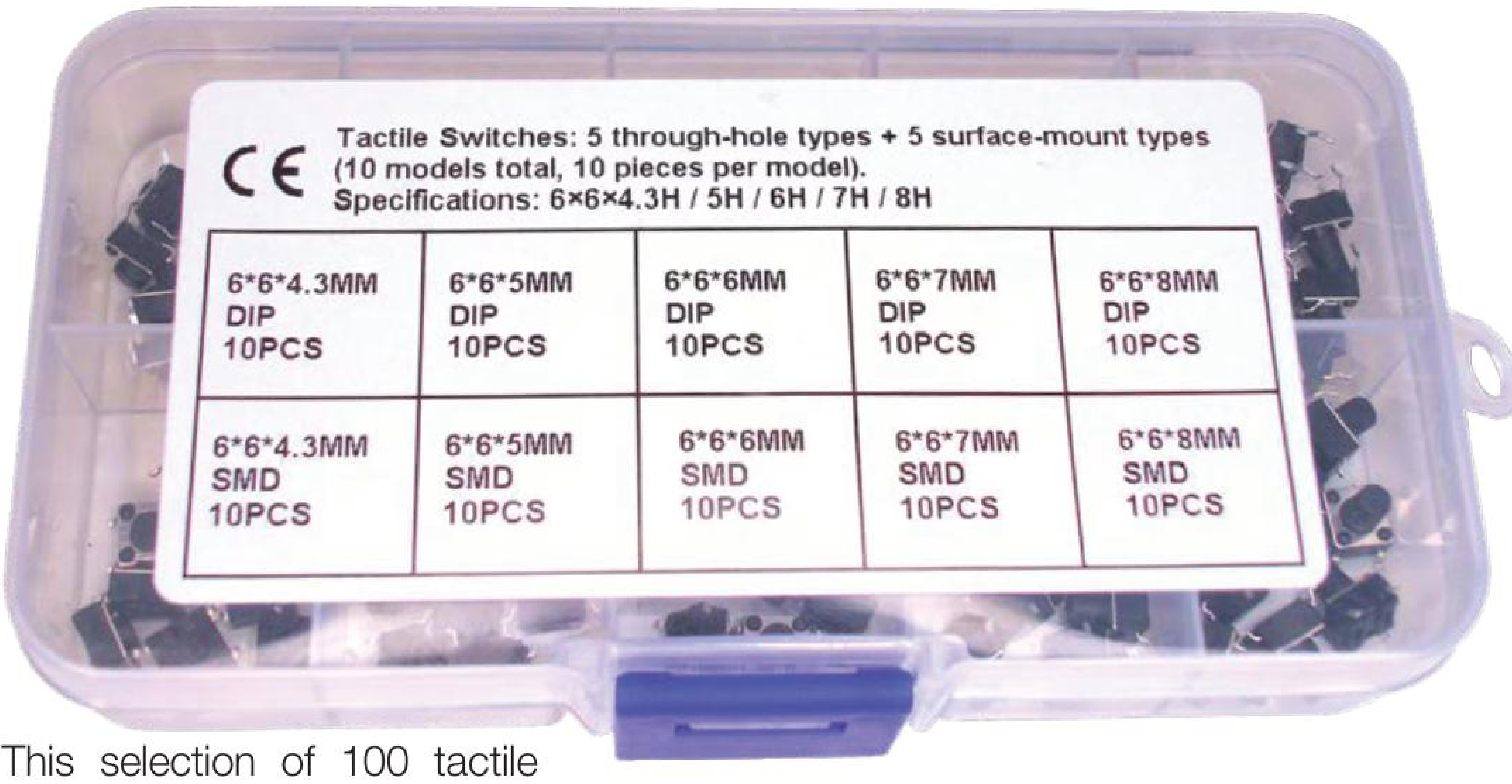
With the dealer taking no responsibility and only giving the usual vague "It's up to you" advice, I plumped for a new EFB battery anyway.

The ultimate upgrade would be a more costly absorbent glass mat (AGM) type, if you can find the right dimensions and battery post arrangement. It's probably best to stick to well-known, reputable brands and check online reviews before buying.

My CTEK MXS 5 charger then suffered a problem with a jammed, unresponsive Mode selector switch. The switch is a thumb-size pushbutton that is likely to get 'mashed' in regular workshop use. My charger was sulking, stuck in motorcycle battery mode.

It appears this is a common problem, as CTEK's own web page suggests "pressing the button a bit firmer and repeat it a few times. If the issue persists, the button is most likely defective. However, please keep in mind that the charger will still work in the selected mode." CTEK's remedy was to suggest 'mashing' the switch even more!

Some users reported that they tried using switch cleaner aerosol, while others resorted to cutting open the front panel and poking the switch internals with a biro. After drilling out a tamper-evident plug, I took my charger apart and discovered that the switch was only a common 6mm PCB tactile pushbutton, of the type found on small electronic gadgets.



This selection of 100 tactile switches from Temu costs £2.

I was left wondering whether these were the best choice for the job; having lots of thumb pressure bearing down on a tiny switch underneath was maybe not what they were designed for. The old English phrase "spoiling the ship for a ha'porth of tar" sprang to mind.

As a possible fix, I've just received from Temu a selection box of tactile 6mm buttons with five different actuator sizes (heights), with both DIP and SMD solder terminals. That's 100 switches for under £2! To make life difficult, CTEK's pushbutton has a bespoke-height actuator with a tiny recess on top, so I'll no doubt have to fabricate or adapt something.

My next project, then, is to see if I can successfully repair my £80 charger with some solder paste and a membrane switch costing one penny. I'll let you know if my efforts pay off!

### Checking out the power

*Net Work* has previously described a range of USB inline monitoring devices, small LCD gadgets that show the voltage, current and other parameters when charging rechargeable products. They are also handy when using USB Power Delivery (PD) compatible products; it's interesting to see the display 'handshaking' as the charging voltage ramps up from a baseline of 5V to deliver more power.

My preference is the low-cost Kowsi KWS X-1 USB-C monitor with multi-colour display (see page 63, December 2025 issue), which I use most of the time. It also offers an oscilloscope-type display with a timebase showing voltage and ripple voltage, amongst other factors.

As many gadgets and apparatus use the classic 5.5mm DC jack, the Kowsi DC-28 (*Net Work*, November 2025) works well with ordinary mains adaptors, as used by my 25V rechargeable vacuum cleaner, for example. A short DC extension lead is handy for hooking up the inline monitor, so I can see at a glance the current state of charge and guess when it's likely to be finished.

Not every idea pays off, though – I also tried one with the PSU for my Asus laptop. Some trial and error revealed that my laptop used a DC jack measuring 4.0 × 1.35mm. To complicate matters, there's a similar 4.0 × 1.7mm type, which didn't fit.

Eventually, I sourced a suitable set of DC adaptors/reducers from AliExpress, but the laptop connections proved dodgy at best, so I gave up on that idea!

Another handy gadget that caught my eye is an inline USB-C voltage adaptor with a difference: it exploits the principle of PD 'triggering' to boost the output voltage from a compatible USB power bank or mains adaptor. The adaptor is equipped with a USB-C input lead, a 7-segment display and a standard 5.5mm DC 1m-long output lead.

Pushbuttons on the device select the DC output voltage. I tested one using several power banks and obtained 9V and 12V DC outputs. It also worked well with a PD-compatible UGREEN mains adaptor; I could derive the advertised options of 5V, 9V, 12V, 15V and 20V from it.

Where a USB-C port isn't available, a USB-C-to-A-type converter can connect the voltage adaptor to traditional-style USB ports instead. These PD adaptors could prove handy for re-purposing a common power bank to supply 9V or 12V DC like a traditional mains adaptor, or you could utilise a PD-compatible mains adaptor to obtain higher voltages (9-20V) for your projects.

They are widely available online for under £3.

### In for the long Haul?

Mention the Chinese website Temu to someone and you'll be greeted either with knowing smiles or hoots of derision. Temu came to Britain quite late in the day and, regardless of one's opinion about the merchandise offered by some sellers, what's undeniable is that Temu has grown exponentially to become one of the world's largest e-commerce platforms.

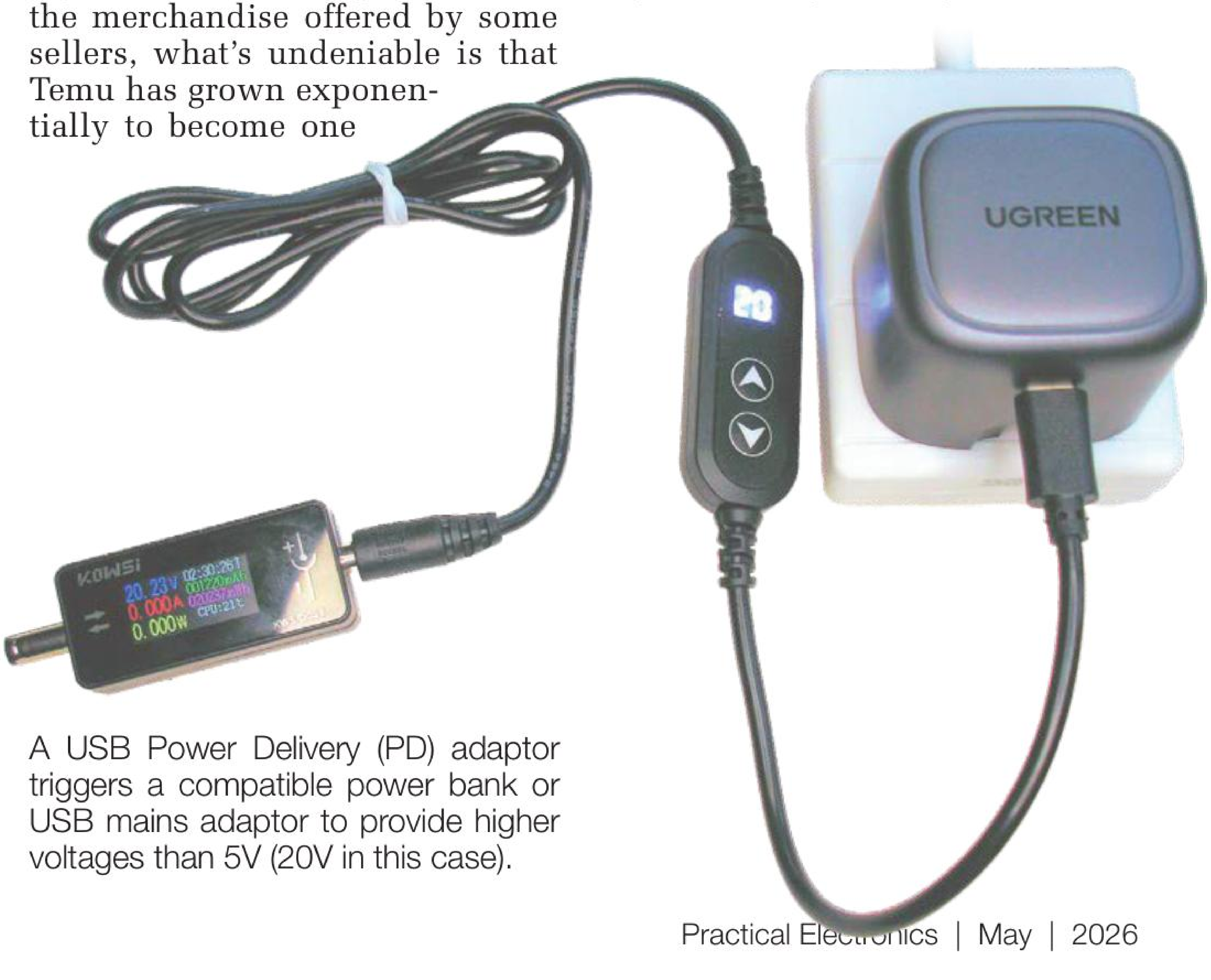
There can be no dispute that Temu's attention to error-free detail at a granular level, especially on the mobile app, is nothing short of sublime. In my view, it makes rivals such as AliExpress appear very clunky and unvarnished in comparison. AliExpress is clearly rattled, because a survey they sent me was obviously fishing for opinions about Temu.

Amazon's answer to Temu is called "Amazon Haul". It lists many low-cost items, but there's just a fraction of the choice offered by Temu. I soon found items on Haul that were three times more expensive than on Temu. Haul is still a work in progress, and how long Amazon bothers with it remains to be seen.

As I wrote in earlier columns, substantial savings can be made on Temu if you know what to look for, and such savings are certainly welcome these days. There can be dubious products of questionable quality; reviews can be insightful. I steer clear of most mains-powered products like unbranded or oddly named electric heaters, steamers, irons, power tools, or similar.

Apart from the odd duff, low-cost purchase, I remain quite defensive of Temu and think it's a mistake to write it off. It's little different from buying through eBay, Amazon, Facebook Marketplace or TikTok Shop, for example.

Not every reader agrees, as my mailbox reminds me, but freedom of choice is a good thing and, in my case, every delivery has arrived



A USB Power Delivery (PD) adaptor triggers a compatible power bank or USB mains adaptor to provide higher voltages than 5V (20V in this case).

as expected, with any problems soon rectified.

Questions about quality and safety apply to goods imported through other platforms, including Amazon Marketplace and AliExpress. The risk of buying items with fake approvals is ever-present. I sourced a decent-looking digital pen-type soldering iron via AliExpress, and when I plugged it in, the tip literally glowed red hot! It was supposedly CE approved and stamped for 220V AC.

A 7000mAh LED rechargeable ceiling light sourced from Amazon had, as I would discover, a Li-ion battery stuck face-down inside – presumably to hide the fact that it was labelled 5000mAh. I couldn't be bothered returning it.

One popular and highly knowledgeable Scottish YouTuber, Big Clive ("the trashiest electronic channel on YouTube") delights in sourcing sundry and sometimes lethal-looking electronics from China and ripping them apart. His "AliExpress Unbagging Joy" videos entertain his 1.1 million subscribers as he reverse-engineers a variety of 'tat' bought online.

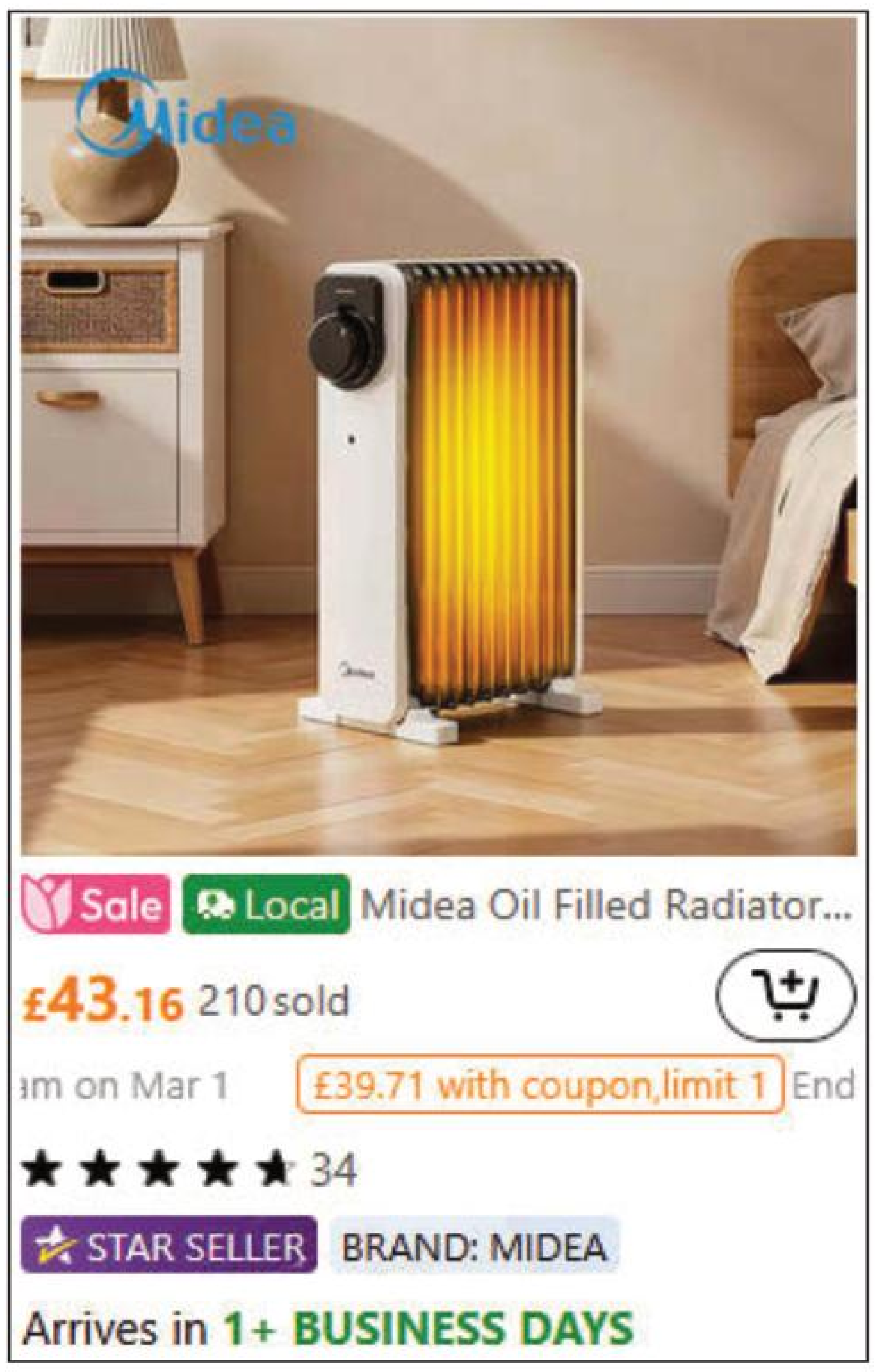
It's not all bad, though; sometimes he's impressed by what he finds.

### The Temu Tax

Buying low-cost goods from Chinese platforms exploits the fact that they land in Britain duty-free, although VAT is levied on top. What has gone unreported so far is last year's Budget announcement by UK Chancellor Rachel Reeves that the *de minimis* rule for imported goods is planned to be eliminated.

The UK duty-free allowance for personal goods is £135 (US $185) and the budget proposals, intended to please British-based businesses and fend off 'unfair' foreign competition, would see import duties applied to all shipments regardless of value.

Charging a nominal import duty (say 2.5-4%) on top may not have much impact on low-value orders. The problem lies in the extra cost of collecting and processing these import duties, which is why I think buyers may eventually see a price hike.


Midea
Sale
Local
Midea Oil Filled Radiator...
£43.16 210 sold
ım on Mar 1
£39.71 with coupon,limit 1
End
★★★★★ 34
STAR SELLER
BRAND: MIDEA
Arrives in 1+ BUSINESS DAYS

'Local warehouse' items are shipped directly from UK warehouses rather than China and arrive in a day or two.

In my industry career, I've done my fair share of importing and exporting (even driving the forklift!). In the world of shipping, the term 'brokerage' covers the work undertaken by import agents to clear loads through customs, sort out the onerous paperwork and collect any import duty before delivering the goods. These admin fees are charged to the importer.

At the consumer level, customs clearance fees can make smaller orders almost non-viable.

Some years ago, late one Sunday night, I ordered some camera equipment online from B&H Photo in New York. My order enjoyed the fastest logistics turnaround I ever saw: the photo gear arrived at my door early on Wednesday morning, and I gave the courier a cheque for the import fees, the goods having arrived in the UK 'delivered duty unpaid' (DDU).

The sting in the tail was that my order had been split into two consignments and therefore across two invoices. This attracted two lots of customs fees, which effectively nearly doubled the delivery cost.

For now, consumers can enjoy buying low-cost merchandise online under the prevailing *de minimis* rules. How the new proposals will play out remains to be seen, but they will eventually affect all imports, including printed circuit boards sourced from China.

Updating systems to recover customs clearance fees from consumers won't happen overnight, and financial sources suggest "by 2029" as the likely timetable.

One emerging trend is 'local warehouse' sales, where independent British-based sellers use the Temu platform to sell merchandise; especially heavier, bulkier lines such as the oil-filled radiator shown in my example.

Goods are sold and shipped from within Britain and usually arrive within a day or two.

There is plenty of evidence that the 'local warehouse' channel is becoming more popular with traders. You can check out the surprisingly wide range of categories on their website, and suppliers can search for "Temu Seller Center" to learn more about selling that way.

Join me next time for more *Net Work*. You can contact the author at alan@epemag.net **PE**

ENIGMA
COMPONENT SHOP
www.enigma-shop.com
Electronic Components for the Hobbyist!
Cable & Wire • Capacitors • Connectors • Filters
Kits & Modules • Hardware & Cooling
*and much more...*
READY TO BE REWARDED?
Join our loyalty scheme and **earn points** on every purchase you make which can be **used for discounts** in the future.
Read our reviews on ★Trustpilot

MAX'S COOL BEANS

By Max the Magnificent

# WEIRD & WONDERFUL ARDUINO PROJECTS

## Part 17: Chronological contraptions and solid snakes

Goodness gracious me; this month, I'm celebrating the 23rd anniversary of the 23rd anniversary of my 23rd birthday. That's not something I expected to hear myself say deep in the mists of time when I was completing my maiden voyage around the sun!

All I can say is, "Where did all those years go?" Well, that's not entirely true; I have lots more to say. Later in this column, we will embark on a new project to create a Curiously Complex Chronological Countdown Contraption to mark the years, months, days, hours, minutes, and seconds until the start of my 100th birthday. But first…

### Variety is the spice of life

We closed last month's column with a four-segment snake randomly meandering around our retro game console's 14 × 10 array of tricolour light-emitting diodes (LEDs). To keep things simple, we started by using our **InitializeSnake()** function to start drawing our snake's head at a specified location in the array, (X, Y) = (7, 5), and move it in a specified direction across the array (**EAST**).

This means that every time we press the reset button on our game console, the snake will respawn at the same spot and start slithering in the same direction.

You can refresh your recollections of this program with the code in the file named **CB-may26-code-01.txt**. As usual, all files mentioned in this column are available as part of the May 2026 download package from the PE website at https://pemag.au/link/acb7

Let's add a bit of visual and user interest by modifying this routine so the snake appears at a random location and starts off moving in a random direction. The new code file is named **CB-may26-code-02.txt**.

Our **InitializeSnake()** function now commences as illustrated in Listing 2(a). We're using the convention that the listing number (2 in this case) corresponds to the numerical part of the matching code file name ("02" here).

```
430  void InitializeSnake ()
431  {
432    uint8_t   tmpX;
433    uint8_t   tmpY;
434    int8_t  deltaX;
435    int8_t  deltaY;
436
437    // Start with a 4-segment snake
438    SnakeHeadPtr    = START_HEAD_PTR;
439    SnakeTailPtr    = START_TAIL_PTR;
440    SnakeLength     = START_LENGTH;
441
442    // Generate random head location and direction
443    tmpX = random(5, 10);
444    tmpY = random(4,  6);
445    SnakeDirection = random(0, NUM_DIRECTIONS);
446
447    // Generate delta values (opposite to direction)
448    if (SnakeDirection == NORTH)
449      {deltaX =  0; deltaY = -1;}
450    else if (SnakeDirection == SOUTH)
451      {deltaX =  0; deltaY =  1;}
452    else if (SnakeDirection == EAST)
453      {deltaX = -1; deltaY =  0;}
454    else if (SnakeDirection == WEST)
455      {deltaX =  1; deltaY =  0;}
```

Listing 2(a): the modified start of the snake initialization routine.

We're starting with a 4-segment snake: one head segment and three body segments. On lines 443 and 444, we use the Arduino's **random()** function to randomly generate the X & Y coordinates for the snake's initial head position such that it will appear somewhere in a small central zone of the display; close enough to the middle to feel fair, but random enough to keep things interesting.

On line 445, we randomly select the direction in which the snake is heading. On lines 448 through 455, we generate the delta values we'll use to build the body of our snake. In this case, we want these values to be opposite to the snake's direction. For example, if the snake starts off moving east, we want its body segments to appear to the west of the head.

The remainder of this function remains unchanged from last month: first generating the snake, loading it into our circular buffer, then drawing it on our array.

### The problem with randomness

There's a classic Dilbert cartoon in which our hero visits the accounting department and is introduced to their "random number generator." The creature assigned to this role is monotonously chanting, "Nine nine nine nine nine nine…", clearly repeating the same number over and over.

Dilbert asks, "Are you sure that's random?" And the manager of the department replies, "That's the problem with randomness. You never *can* be sure."

With our current Snake program, if you look closely, you'll see that every time we press the reset button, the snake appears in the same "random" position and moves in the same "random" direction. Hmm, that's perhaps not quite as "random" as we might have been expecting.

Generating a truly random number is a lot harder than one might think. The Arduino Uno (or Nano) does not generate truly random numbers. Instead, it generates pseudo-random numbers. Although these appear to be statistically random, they are actually produced by a deterministic mathematical algorithm called a pseudo-random number generator (PRNG).

This means that if we start the algorithm with the same initial ('seed') value, it will always produce the same sequence of "random" numbers.

We can easily verify this using the simple test program shown in Listing 3(a), from the file named **CB-may26-code-03.txt**. On line 5, we use the Arduino's built-in randomSeed() function to specify the seed value that will be used by the random() function.

Remember to open the serial monitor window in the Arduino's integrated development environment (IDE) so you can see what's happening, then upload this program into your Arduino. As you'll see, we get exactly the same six-number sequence (38, 52, 78, 17, 41, 58) every time we reset the console and rerun the program.

One point of interest here is that good pseudo-random number generators exhibit a strong sensitivity to initial conditions, meaning that even a one-bit change in the seed produces a completely different output sequence.

Try this for yourself by changing our seed value from 1234 (10011010010 in binary) to 1235 (10011010011 in binary), which involves flipping only the least significant bit, and compare the new sequence (45, 1, 4, 28, 71, 30) to the old one.

On a standard Arduino (eg, Uno or Nano), the default seed value is 1. More precisely, if we don't call the randomSeed() function ourselves, then the internal PRNG is initialised as though we had called randomSeed(1). This is why, unless we intervene, every "random" game begins in exactly the same way.

In many embedded applications, pseudo-random values are perfectly adequate; repeatability can sometimes help with debugging. In other cases, however, we may want things to appear to be a tad more random. A cheap-and-cheerful approach to increasing the perceived randomness is illustrated in Listing 4(a), from the file named **CB-may26-code-04.txt**.

All we've done is modify the way we specify the seed value on line 5. If analog input A0 is left floating (unconnected), it will pick up electrical noise. That means the voltage on this pin will vary each time we read it, yielding a different seed and, hence, a different sequence.

Try running this new version of our test code several times and observe that the generated sequence differs each time you upload the program and each time you press the reset button on the console.

I've just added this analog-based random seed generation to the setup() function in our Snake program (**CB-may26-code-05.txt**). Now, whenever we press the reset button to start a new game, the snake really does appear at a random location heading in a random direction (well, random enough for us).

### Taking control

Now, it's time to take our snake's reins in our hands, which is another thing I didn't expect to hear myself saying when I woke up this morning.

You might recall that our current UpdateSnake() function starts as illustrated in Listing 5(a). The key part to observe here is that, until now, using lines 497 through 502, we've been randomly generating the snake's new direction. After this, the function generates the next X/Y location for the head, removes the tail segment, draws the new head segment, and so forth.

Since we now plan to take control ourselves, we can remove lines 497 through 502 and line 486, which we no longer need.

All that remains is to bring our direction control switches into play. Remember that our loop() function repeatedly calls UpdateSnake(), which updates the snake's position on the array, UpdateSwitches(), which debounces our switches, and ProcessSwitches(), which performs any game-related actions associated with the switches being pressed or released.

What we are going to do is tweak the ProcessSwitches() routine so that pressing any of the north, south, east, or west buttons updates the SnakeDirection variable. It's this new value that the UpdateSnake() function will use the next time it's called.

The modified code for the north switch in the ProcessSwitches() routine is shown in Listing 6(a) (**CB-may26-code-06.txt**). The only change occurs on line 267, which we've added to set the snake's direction to NORTH when the north switch is pressed (goes active). Similar changes are required for the south, east, and west switches.

```
1   void setup()
2   {
3     Serial.begin(9600);
4
5     randomSeed(1234);
6
7     Serial.println();
8     Serial.println("======");
9     for (int i = 0; i < 6; i++)
10    {
11      Serial.println(random(100));
12    }
13  }
```

Listing 3(a): using a fixed random seed.

```
1   void setup()
2   {
3     Serial.begin(9600);
4
5     randomSeed(analogRead(A0));
6
7     Serial.println();
8     Serial.println("======");
9     for (int i = 0; i < 6; i++)
10    {
11      Serial.println(random(100));
12    }
13  }
```

Listing 4(a): a more random random seed.

```
483 void InitializeSnake ()
484 {
485   uint8_t tmpSnakeHeadPtr;
486   uint8_t tmpDirection;
487   uint8_t   tmpX;
488   uint8_t   tmpY;
489   int8_t  deltaX;
490   int8_t  deltaY;
491
492   if ( (TimeNow - SnakeThen) >= SNAKE_MOVE_TICKS)
493   {
494     // Pointer for new head in circular buffer
495     tmpSnakeHeadPtr = (SnakeHeadPtr+1) % NUM_SEGMENTS;
496
497     // Pick a new direction (can't be 180 degrees)
498     do {
499       tmpDirection = random(0, NUM_DIRECTIONS);
500     } while (tmpDirection == (SnakeDirection ^ 1));
501
502     SnakeDirection = tmpDirection;
```

Listing 5(a). automatically generating the snake's direction.

```
260   // Switch North
261   if (Switches[SW_NORTH].swState == JUST_GONE_ACTIVE)
262   {
263     // Mandatory
264     Switches[SW_NORTH].swState = WAITING_FOR_ALL_ONES;
265
266     // Optional (game-specific)
267     SnakeDirection = NORTH;
268     // Serial.println("Switch North just went active");
269   }
270   else if (Switches[SW_NORTH].swState == JUST_GONE_INACTIVE)
271   {
272     // Mandatory
273     Switches[SW_NORTH].swState = WAITING_FOR_ALL_ZEROS;
274
275     // Optional (game-specific)
276     // Serial.println("Switch North just went inactive");
277   }
```

**Added line (similar lines added for the other three switches)**

Listing 6(a): tweaking the switch-handling routine.

Try running the latest version of our program and bask in the glow of controlling your snake.

### But what about…

I'm afraid that there's a fly in the soup and an elephant in the room (I never metaphor I didn't like). Did you spot the obvious omission with respect to the latest and greatest version of our code?

Previously, when our snake was wending its way around the array under its own steam, we ensured it didn't perform an 'about-face' (ie, a 180° change in direction). For example, if the snake was heading east, we allowed it to keep on heading east, or change to moving north or south, but not west.

Now that we are controlling the snake ourselves, we have no such safeguard. So, what will happen if the snake is heading east and we press the west button? Why don't you look at the code to see if you can predict what will occur, then run the program and perform this test to see if what takes place is what you expected?

As you'll see, if we perform a 180° about-face, the snake simply passes through itself, which seems a little unrealistic. At this point, we have three main options. First, we could leave things as-is, which many players would find logically and aesthetically unpleasing. Second, we could cause the snake to 'die' and end this level of the game, which many players would consider overkill (no pun intended).

The third option, which was adopted by many classic Snake implementations, was to simply ignore any attempt by the user to reverse direction, so that's what we'll do.

Implementing this is easy-peasy, lemon squeezy. All we have to do is slightly refine our existing switch processing code in the **ProcessSwitches()** routine. The modified code for the north switch is shown in Listing 7(a) (**CB-may26-code-07.txt**).

All we've done is add a check on line 267 to ensure that, when the north button is pressed, we only set the snake's direction to **NORTH** if it's not already going **SOUTH**. Corresponding changes are required for the south, east and west switches.

```
260   // Switch North
261   if (Switches[SW_NORTH].swState == JUST_GONE_ACTIVE)
262   {
263     // Mandatory
264     Switches[SW_NORTH].swState = WAITING_FOR_ALL_ONES;
265
266     // Optional (game-specific)
267     if (SnakeDirection != SOUTH) SnakeDirection = NORTH;
268     // Serial.println("Switch North just went active");
269   }
270   else if (Switches[SW_NORTH].swState == JUST_GONE_INACTIVE)
271   {
272     // Mandatory
273     Switches[SW_NORTH].swState = WAITING_FOR_ALL_ZEROS;
274
275     // Optional (game-specific)
276     // Serial.println("Switch North just went inactive");
277   }
```

Listing 7(a): the refined switch-handling routine.

```
// Set the color of pixel at [x, y] (i.e., [column, row])
void PokePixel (uint8_t x, uint8_t y, uint32_t color)
{
  uint8_t n = GetNeoNum(x, y);
  Neos.setPixelColor(n, color);
}

// Read the color of pixel at [x, y] (i.e., [column, row])
uint32_t PeekPixel (uint8_t x, uint8_t y)
{
  uint8_t n = GetNeoNum(x, y);
  return(Neos.getPixelColor(n));
}
```

Listing 8(a): peeking and poking pixels.

```
537 void PlaceFood()
538 {
539   uint8_t tmpX;
540   uint8_t tmpY;
541
542   do {
543     tmpX = random(0, NUM_COLS);
544     tmpY = random(0, NUM_ROWS);
545   } while (PeekPixel(tmpX, tmpY) != EMPTY_COLOR);
546
547   PokePixel(tmpX, tmpY, FOOD_COLOR);
548 }
```

Listing 8(b): placing a random piece of food.

### PEEK and POKE

On classic 8-bit machines like the Apple ][, Commodore 64, BBC Micro etc, memory wasn't abstracted away. Hardware registers, screen memory, sound chips and system flags all lived at known memory addresses.

Also, on most early computers, the BASIC language (Beginner's All-purpose Symbolic Instruction Code) wasn't just a programming language; it was the operating environment. When you powered the machine on, you were dropped straight into the BASIC prompt.

Two foundational, ubiquitous and slightly dangerous BASIC commands are **PEEK()** and **POKE()**. **PEEK(address)** returns the numeric value stored at a given memory location, while **POKE(address, value)** writes a numeric value directly into the specified memory location.

Currently, we use our **DrawPixel()** function to set (write) the colour of a pixel at a specified (x, y) location on the array. Thus far, we haven't felt the need to be able to access (read) the colour of a pixel, but that's about to change.

As a first step, I've decided to rename our **DrawPixel()** function to **PokePixel()** (see **CB-may26-code-08.txt**). If you are modifying your own code, I suggest changing the function name and all calls to it, then verifying that everything still works by recompiling before proceeding further.

Next, I've added a new **PeekPixel()** function. Both of these functions are shown in Listing 8(a).

### Ooh, tasty!

A core concept of the Snake game is that the snake grows larger as it munches on tasty morsels of 'food'. As usual, we'll take this step-by-step. We'll start by defining **EMPTY_COLOR** as **BLACK** and **FOOD_COLOR** as **YELLOW** (or any colour of your choice, remembering that we've already defined **HEAD_COLOR** as **MAGENTA** and **BODY_COLOR** as **GREEN**).

Next, we'll create a utility function called **PlaceFood()** that places a pixel of 'food' at a randomly selected unoccupied location. I toyed with the idea of calling this function **ServeSnack()**, but I decided that would be super silly (as you must already know, I'm a very serious person). This function is shown in Listing 8(b).

On line 542, we initiate a **do** loop. On lines 543 and 544, we generate random X and Y values. We keep on

```
577 void DealWithDeadSnake ()
578 {
579   uint8_t tmpPtr = SnakeHeadPtr;
580
581   // Color the snake red
582   delay(1000);
583   for (uint8_t i = 0; i < SnakeLength; i++)
584   {
585     PokePixel(Snake[tmpPtr].x, Snake[tmpPtr].y, DEAD_COLOR);
586     tmpPtr = (tmpPtr + NUM_SEGMENTS - 1) % NUM_SEGMENTS;
587     Neos.show();
588     delay(100);
589   }
590   delay(1400);
591
592   // Start a new game
593   InitializeSnake();    // Generate and draw snake
594   InitializeDisplays(); // 7-Segment displays
595 }
```

Listing 9(a): snake funeral arrangements.

doing so until we find an unoccupied pixel using the test on line 545.

I should point out that this is far from being the most efficient way to achieve this functionality, but it does give us an opportunity to take our new **PeekPixel()** and **PokePixel()** functions for a spin.

One more minor modification is to call our new **PlaceFood()** function at the end of the **InitializeSnake()** routine. We do this just after we've generated our initial snake and drawn it on the array.

Try running this new version of the program and observe that every time you press the console's reset button, both the snake and the yellow food pixel appear at random locations on the array.

### Feed me!

At the moment, our snake doesn't know what to do with any food it encounters. If we use our direction controls to guide the snake to the food, it simply tramples it underfoot (you know what I mean). So, the next step is to modify our **UpdateSnake()** function so it can 'see' and 'eat' the food.

To be honest, this takes a bit of mental gymnastics to wrap one's head around. The way the **UpdateSnake()** function currently works is as follows:

1. Based on the present location and direction of the snake's head segment, determine the X and Y array values for the new head segment location.
2. Remove the old tail segment.
3. Set the old head segment's location to the snake's body colour.
4. Set the new head segment's location to the snake's head colour.

As we discussed last month, the reason we remove the old tail segment **before** drawing the new head segment, thereby allowing the head to move into the square the tail has just vacated, is that this reflects the behaviour of the classic Snake game.

By removing the tail first, we avoid falsely detecting a collision with the tail-end of the snake's own body and allow the familiar smooth, continuous motion players expect. The thing is that if the snake's head moves into a pixel containing food and eats it, the way the snake grows is by **not** removing the old tail segment. This means that we need to modify the flow as follows:

1. Based on the present location and direction of the snake's head segment, determine the X and Y array values for the new head segment location.
2. If the new head segment location contains food, don't delete the tail segment; otherwise, delete it.
3. Take note of whatever is in the new head segment location: options are nothing (empty), food, or a segment of the snake's body.
4. Set the old head segment to the snake's body colour.
5. Set the new head segment to the snake's head colour.
6. Perform appropriate actions according to what was in the new head segment location before we moved the head.

Regarding point 6, we now have three options:

i. If the location was empty, then there's nothing left for us to do.
ii. If the location contained food, we need to increment and display the new score (ie, the new length of the snake). We also need to call our **PlaceFood()** function to present a new pixel snack.
iii. If the location contained one of the snake's body segments, we set a status flag to say that the snake just died (we'll handle this outside of this routine).

I've just implemented all of these changes in the file named **CB-may26-code-09.txt**. I have also added a global variable called **SnakeStatus**. This starts off being set to **SNAKE_IS_ALIVE**. When the player causes the snake to run into itself, the **UpdateSnake()** function sets this flag to **SNAKE_JUST_DIED**.

I've also modified the end of the **loop()** function. After calling **UpdateSnake()**, we test whether the snake has just died. If so, we call a new **DealWithDeadSnake()** function, as illustrated in Listing 9(a).

You can make your version as fancy as you wish, or you can keep it 'mean and lean'. In my case, I opted to pause for a second to let me reflect on my mistake, then colour the snake's segments red, starting with the head and ending with the tail, with a 1/10th of a second (100ms) delay between each segment.

Finally, on lines 593 and 597, we reinitialise the snake and the 7-segment displays, after which we'll start the game anew. I'm running the latest version of the program as we speak, and it's starting to look very tasty indeed.

### It's the final countdown (do do do dooo)

I suddenly find *The Final Countdown*, the 1986 hit by the Swedish rock band Europe, rattling around my noggin.

This is because I've decided to tweak our **InitializeDisplays()** function. At the moment, this kicks off by displaying the game level (we're starting at one, but I envisage we'll eventually support nine levels) and the length of our starting snake (currently four):

**L1 □0 04**

We are using the three least significant digits to display the snake's length, which has a maximum possible value of 140. I'm using the □ character above to represent a blank, unlit digit on the 7-segment displays.

What we're going to do is add a countdown before the game (or a new level) starts. In the case of our current L1 implementation, this will look like the following on the 7-segment displays:

**55 55 55**
**44 44 44**
**33 33 33**
**22 22 22**
**11 11 11**
**L1 □0 04**

I've just added this to our code (in the file named **CB-may26-code-10.txt**) and I think it looks great. Of course, you may decide to omit this completely or to do it differently. You're in charge of your own console and the master or mistress of your own destiny (just don't let this power run to your head).

### The world is our Oyster Kilpatrick

At this point, we have so many options open to us that I barely know where to start. One thing we could do is define the characteristics of the first three levels: L1, L2, and L3, and what it takes to progress from one level to the next.

For example, in L1, we currently update the snake's position every 500ms (twice per second). We could reduce this to 400ms in L2 and 300ms in L3. We could also say that if we are in L1, we automatically proceed to L2 when the snake reaches 30% of its maximum possible length (140), which would be 42 segments. If the snake dies before we reach this magic number, we have to replay the level.

Similarly, if we are in L2, we could say that we need the snake to reach 40% of its maximum length before proceeding to L3. And if we are in L3, the snake must reach 50% of its maximum length to progress to L4.

Levels L4, L5, and L6 could be identical to levels L1, L2, and L3, respectively, except that the food only persists at its current location for a random number of cycles before disappearing and reappearing elsewhere in the array.

In the same vein, levels L7, L8, and L9 could be identical to levels L4, L5, and L6, except that, in addition to regular food pixels (**YELLOW**), we are also occasionally presented with poisoned food pixels (**CYAN**), the eating of which will cause the snake to slither off this mortal coil.

That seems appropriate given that cyanide (HCN) is quite poisonous.

Anyway, the suggestions above are just "top of my head" ideas. You can construct your levels any way you wish. The world is truly your oyster (or sea creature of your choice).

There are many other possibilities. Some are aesthetic, like making the food and poison pixels flash, as we did in our Etch A Sketch program. Others are more functional, such as removing the wraparound capability and making the array's walls (edges) 'solid'.

This would present a new choice: should the snake die if it runs head-first into a wall, or should it simply stop moving and wait for the user to turn its head before setting off again?

More ideas are popping into my head, like having occasional 'special' food pixels (**GREEN**) appear that, if eaten, make the snake grow by two or three segments at a time. We could also keep track of how many moves the snake makes between eating food particles. If the player takes too long, the score (and the snake's length) could start to shrink.

I should probably point out that I won't be adding any of the aforementioned functionality in future columns, because we have bigger fish to fry. Having said this, I may occasionally surprise you by writing something like, "You can download a new version of the snake program that…"

| Binary $2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$ | Dec | Hex |
|---|---|---|
| 0 0 0 0 0 0 0 0 | +0 | 0 0 |
| 0 0 0 0 0 0 0 1 | +1 | 0 1 |
| 0 0 0 0 0 0 1 0 | +2 | 0 2 |
| 0 0 0 0 0 0 1 1 | +3 | 0 3 |
| : | : | : |
| 0 1 1 1 1 1 0 1 | +125 | 7 D |
| 0 1 1 1 1 1 1 0 | +126 | 7 E |
| 0 1 1 1 1 1 1 1 | +127 | 7 F |
| 1 0 0 0 0 0 0 0 | +128 | 8 0 |
| 1 0 0 0 0 0 0 1 | +129 | 8 1 |
| 1 0 0 0 0 0 1 0 | +130 | 8 2 |
| : | : | : |
| 1 1 1 1 1 1 0 0 | +252 | F C |
| 1 1 1 1 1 1 0 1 | +253 | F D |
| 1 1 1 1 1 1 1 0 | +254 | F E |
| 1 1 1 1 1 1 1 1 | +255 | F F |

(a) Unsigned binary

| Binary $-2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$ | Decimal | Hex |
|---|---|---|
| 0 0 0 0 0 0 0 0 | +0 | 0 0 |
| 0 0 0 0 0 0 0 1 | +1 | 0 1 |
| 0 0 0 0 0 0 1 0 | +2 | 0 2 |
| 0 0 0 0 0 0 1 1 | +3 | 0 3 |
| : | : | : |
| 0 1 1 1 1 1 0 1 | +125 | 7 D |
| 0 1 1 1 1 1 1 0 | +126 | 7 E |
| 0 1 1 1 1 1 1 1 | +127 | 7 F |
| 1 0 0 0 0 0 0 0 | –128 | 8 0 |
| 1 0 0 0 0 0 0 1 | –128 + 1 = –127 | 8 1 |
| 1 0 0 0 0 0 1 0 | –128 + 2 = –126 | 8 2 |
| : | : | : |
| 1 1 1 1 1 1 0 0 | –128 + 124 = –4 | F C |
| 1 1 1 1 1 1 0 1 | –128 + 125 = –3 | F D |
| 1 1 1 1 1 1 1 0 | –128 + 126 = –2 | F E |
| 1 1 1 1 1 1 1 1 | –128 + 127 = –1 | F F |

(b) Signed binary

Fig.1: unsigned vs signed binary representations.

### The sound of silence

I love the song *The Sound(s) of Silence* by Simon and Garfunkel (oddly, its title was plural on the album but became singular for the single release). My point is that, while playing the current version of my snake game, I realised that the sound (or lack thereof) was deafening.

It struck me that this game, and others to come, would be greatly enhanced by the addition of some simple sound effects, like a cheerful 'cheep' when the snake eats food, or a mournful melody when the snake dies, for example. The more I think about this, the more I want to do it, so that's one of the things we will implement next month.

### A sign of the times

Way back in the mists of time (the January 2026 issue of *PE*, to be more precise), we introduced the concepts of logical and arithmetic shifts. At that time, I wrote, "The reason we might prefer an arithmetic shift right over a logical shift right with signed integers is too involved to go into here (we'll discuss this further in our next column)".

Sad to relate, we never did that. Well, it's time to stop "prevaricating around the bush", as Wallace might say.

The differences between unsigned and signed integers, and the many ramifications that flow from them, can be confusing to the uninitiated. But the deeper we dig into tasks like bit manipulation (eg, for use with the games for our console), the more we need to understand what's going on; otherwise, many things won't make much sense.

For the sake of simplicity, we'll focus on eight-bit values. Let's start with the fact that eight bits can be used to represent $2^8 = 256$ different combinations of 0s and 1s. We can use these patterns to represent anything we wish, including numbers, letters, and symbols.

In the case of a variable of type **uint8_t** (unsigned 8-bit integer), we can represent positive decimal (dec) values in the range 0 to +255, as illustrated in Fig.1(a). These correspond to hexadecimal (hex) values from 00 to FF.

Now suppose we wish to represent both positive and negative values. There are several ways we could do this, but the technique used by today's computers is commonly referred to as 'signed binary', although the formal terminology is "two's complement representation" (we'll look into one's complement and two's complement representations more next month).

In the case of a variable of type **int8_t** (signed 8-bit integer), we can represent values in the range -128 to +127, as illustrated in Fig.1(b). These still correspond to hexadecimal (hex) values in the range 00 to FF, although there is nothing stopping you from thinking of half that range as negative hexadecimal numbers, if you prefer.

The signed binary representation can be tricky to wrap your brain around the first time you see it but it's actually amazingly efficient and efficacious. The key concept is that the most significant bit (bit 7, in the case of an 8-bit value) doesn't represent only the sign

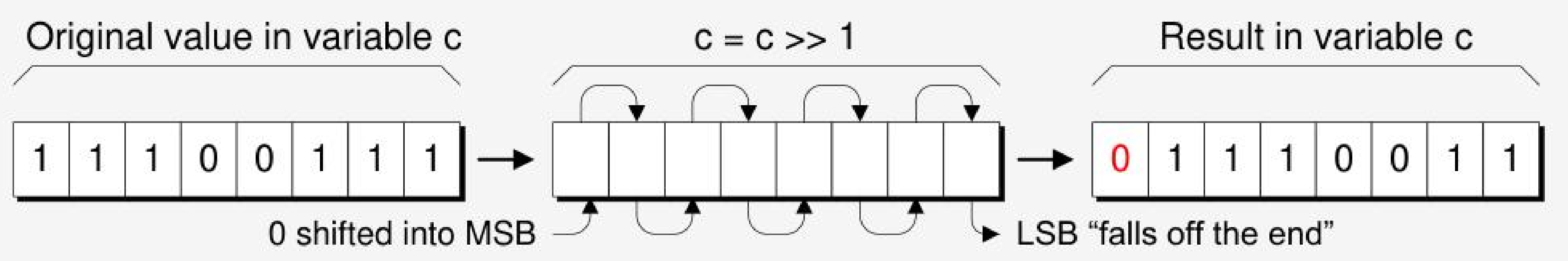
```
uint8_t uA = 204;
int8_t  sB =  -1;
uint8_t uY = uA + sB;

  1 1 0 0 1 1 0 0  (uA = 204)
+ 1 1 1 1 1 1 1 1  (sB =  –1)
= 1 1 0 0 1 0 1 1  (uY = 203)
```

(a) Unsigned + signed

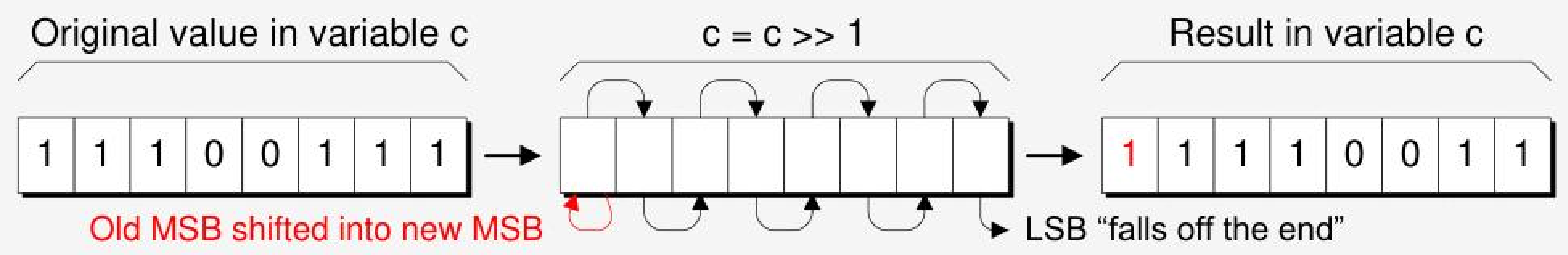
```
int8_t sA = -52;
int8_t sB =  -1;
int8_t sY = sA + sB;

  1 1 0 0 1 1 0 0  (sA = –52)
+ 1 1 1 1 1 1 1 1  (sB =  –1)
= 1 1 0 0 1 0 1 1  (uY = –53)
```

(b) Signed + signed

Fig.2: adding unsigned and signed integers.



(a) logical shift right on signed 8-bit value (int8_t)



(b) arithmetic shift right on signed value (int8_t)

Fig.3: logical versus arithmetic right shifts.

of the number; instead, it represents a negative quantity (-128, in the case of an 8-bit value).

The remaining bits continue to represent their regular positive values.

Although this may, at first glance, appear to be an outrageously complicated solution to a seemingly simple problem, there's method to its madness. For example, exactly the same underlying operations are used by the computer to add and subtract unsigned and signed integers.

Furthermore, it's possible to 'mix and match', adding and subtracting one type to/from another, just so long as we're confident the result will fall within the valid range of the destination variable.

For example, we can add or subtract a variable of type `int8_t` (where this variable may contain a positive or negative value) to a variable of type `uint8_t`, just so long as we know that the result won't be less than 0.

In Fig.2(a), we add an unsigned integer containing +204 to a signed integer containing -1 and store the result in another unsigned integer. Not surprisingly, 204 + -1 equals 203. By comparison, in Fig.2(b), we add a signed integer containing -52 to another signed integer containing -1. In this case, we aren't surprised to discover that -52 + -1 = -53.

The real point here is that the bit patterns of the variables and the operations performed are identical in both cases. The only difference is in how we interpret what these bit patterns represent.

You may recall that we've already seen examples of Fig.2(a) in our console code. Consider Listings 2(a) and 5(a). In both cases, we declare `tmpX` and `tmpY` as `uint8_t`, and `deltaX` and `deltaY` as `int8_t`. Later, we add `deltaX` to `tmpX` and `deltaY` to `tmpY`.

One important point worth noting is that it's usually a good idea to use signed integer variables as `for()` loop control variables. Suppose you get into the habit of using unsigned integers as your control variables; for example, `for (uint8_t i = 0; i < 10; i++) {}`. This will work perfectly well.

The problem only arises when you subsequently implement a loop like `for (uint8_t i = 10; i >= 0; i--) {}`. Because the control variable is unsigned, it can never be less than 0. When `i` reaches 0 and is decremented, it wraps around to 255, and the loop continues forever.

### Ooh, shifty!

The bitwise logical operations `&`, `|`, `^`, and `~` (AND, OR, XOR and NOT, respectively) do not care about numeric interpretations; they simply manipulate the underlying bit patterns. That means they operate on signed and unsigned integers in exactly the same way at the bit level.

In most CPU conversations, shift instructions are grouped with logical (bitwise) instructions because they operate on bits rather than performing arithmetic in the mathematical sense. However, there's a wrinkle when it comes to shifts.

For the sake of simplicity, let's assume we're working with single-bit shifts. In practice, left shifts behave the same at the bit level for signed and unsigned integers, provided we remain within the representable range. In this case, a 0 is shifted into the least significant bit (LSB) and the most significant bit (MSB) 'falls off the end'. This is known as a 'logical shift'.

A right shift performed on an unsigned integer also acts as a logical shift. In this case, a 0 is shifted into the MSB and the LSB 'falls off the end'. However, things get trickier when performing a right shift on a signed integer. In this case, there are two possibilities.

The first is to perform a logical shift, in which case a 0 will be copied into the MSB, as illustrated in Fig.3(a). The second is to perform an arithmetic shift, in which case the original MSB (0 or 1) will be copied into the new MSB, as illustrated in Fig.3(b).

The problem is that the silly sausages who created the original C (and later C++) specifications left this behaviour implementation-defined, meaning the creators of the compiler must document their choice, but the standard does not require a specific one.

Most modern compilers do the logical thing (no pun intended) and perform arithmetic right shifts on signed integers but that isn't mandatory. So we cannot rely on the result, especially if we port our code to a different compiler.

Why does this matter? Early microprocessors and microcontrollers didn't have hardware multipliers or dividers, so these operations were implemented using many simple instructions and many clock cycles. Savvy programmers knew that they could implement certain multiplication and division operations using shift instructions.

Let's start with some unsigned integer examples. Suppose we have a `uint8_t` value of 00001100 in binary (12 in decimal). Shifting our original value left by one bit gives 00011000 (24 in decimal), which is equivalent to multiplying it by two. Similarly, shifting it by two bits gives 00110000 (48 in decimal), which is equivalent to multiplying it by four.

Similarly, shifting our original value right by one bit gives 00000110 (6 in decimal), which is equivalent to dividing by two, while shifting it right by two bits gives 00000011 (3 in decimal), which is equivalent to dividing it by four.

The upshot of all this is that multiplications or divisions by powers of two can be performed in a single clock cycle. There are lots of variations on this. For example, a common requirement

I still have no idea why they call me "Mad Max"!

is to multiply by ten, and N * 10 is the same as (N * 8) + (N * 2), which is the same as (N << 3) + (N << 1).

This means that multiplying an 8-bit integer by ten can be performed using two shifts and one addition, consuming only three clock cycles. As we've previously discussed, in the absence of a hardware multiplier, modern compilers will do this for us, but it's still nice to know what's going on 'under the hood' (or bonnet, if you prefer).

Things also work as expected with signed integers when shifting positive or negative numbers to the left (assuming we don't exceed the range of values that can be supported).

Suppose we have an int8_t value of 11101100 binary (-128 + 108 = -20). Shifting this value by one bit to the left yields 11011000 (-128 + 88 = -40), which is equivalent to multiplying by two, while shifting it two bits to the left will result in 10110000 (-128 + 48 = -80), which is equivalent to multiplying by four.

Now this is where things get tricky. This will also work with signed integers when shifting positive or negative numbers to the right, just so long as the compiler implements an arithmetic shift. For example, shifting our original value one bit to the right yields 11110110 (-128 + 118 = -10), which is the same as dividing by two.

However, if the compiler implements this as a logical shift, we'll end up with 01110110, which is equivalent to +118, which isn't what we want at all. The bottom line is that with unsigned values, shifts are obedient, but with signed values, they demand supervision.

### Chronological contraptions

As noted earlier, we are poised to embark on a new project to create a Chronological Contraption to mark the years, months, days, hours, minutes, and seconds until the start of my 100th birthday celebrations. Of course, you can use it to count down to anything you want. It can also act as a clock.

As usual, I'm collaborating on this project with my friend Joe Farr, because 'two heads are better than one' when it comes to making decisions and designing things. Also, it's more fun to have someone to bounce ideas off. For this project, we've decided to use Soviet-era vacuum fluorescent display (VFD) tubes (Photo 1).

We can think of VFDs as something of a cross between traditional vacuum tubes and Nixie displays. Like tubes, VFDs use thermionic emission within a vacuum envelope, and, like Nixies, they display glowing digits behind glass. The key difference is that VFDs use phosphors rather than gas discharge and operate at far lower voltages.

Photo 1: a vacuum fluorescent display (VFD) inside a glass tube. They also come in multi-digit rectangular assemblies. Source: https://display-tubes.org/vfd/reflector-iv-12/

VFDs are attractive for this project because they work well over a wide temperature range, offer excellent brightness and viewing angle, and have that distinctive 'retro-tech' aesthetic beloved by your humble narrator.

The IV-6 and IV-11 versions of these tubes are readily available as surplus NOS (new old stock) parts from hobby electronics sellers and on eBay, where IV (ИВ in Cyrillic) stands for "Indicator, Vacuum-fluorescent" (or "Indikator Vakuumny" in Russian).

I opted for the IV-11 model/type because these are bigger, and bigger is better. Having said this, these tubes are still only around two inches (50mm) tall and 0.8-0.9 inches (about 20-22mm) in diameter.

I'm using four digits to represent the year and two digits each to represent each of the remaining five fields, so 14 digits in all. I've purchased 20 tubes, just in case.

However, I understand that not all readers will want to undertake a 'full Monty' implementation, so we've come up with a modular design in which each tube sits on its own circuit board. You can connect as many (or as few) of these boards together as you wish. We've also designed small spacer boards to separate the various fields.

### Next time

In addition to adding sound effects to our retro games console, we'll delve deeper into the design of the Chronological Contraption's circuits and circuit boards.

As always, if you have any thoughts you'd care to share on anything you've read here, please feel free to email me at max@clivemaxfield.com

**5-year collections**

**2019-2023**
**£49.95**

**2018-2022**
**£49.95**

**2017-2021**
**£49.95**

**2016-2020**
**£44.95**

Purchase and download at:
www.electronpublishing.com

# The **Fox** Report

Barry Fox's technology column

## Freely's success will depend on good hardware

**Freely is the key to switching** the UK from traditional, linear TV broadcasting to IP/Internet TV. But what is Freely? The official Freely website (**www.freely.co.uk/how-freely-works**) is aimed at those with a short attention span, with next to no nitty-gritty:

*Less scrrrrrrrrolling, more streaming. No more getting your detective on to track down your next fave across different apps. Save loads of time by finding everything in one place. Use the new MiniGuide to whizz from the latest season of The Traitors, to Love Island, or even The Great British Bake Off. If you needed an excuse to get out of going out, it's this!*

I have done my best to get more details and noted the obstacles, if only for easy reference if – I fear when – things go wrong. Although Freely is clever, there looks to be plenty of room for that.

The technology bundles Freeview Digital Terrestrial broadcast TV channels with a selection of streamed channels. The company behind the technology is Everyone TV, until 2023 known as Digital UK. This is a joint venture owned and supported by the UK's leading public service broadcasters: the BBC, ITV, Channel 4 and Channel 5.

Some 20 years ago, Digital UK handled the huge tasks of digital switchover and launching Freeview from digital terrestrial masts and Freesat, the same satellites used by Sky for pay viewing.

At that time, Digital UK saw the value of providing technical journalists with solid facts on the issues and answers to technical questions. So Digital UK employed an ex-technical journalist who understood what technical journalists needed to know and where to find technically solid answers to questions.

The switchover went more smoothly than anyone dared hope, and David Scott, then CEO of Digital UK, was justly knighted in 2013.

Everyone/Freely launched in 2024 with a speech by current CEO Jonathan Thompson at the Digital Television Group (DTG) Conference and now describes itself as "the organisation leading the evolution of free TV in the UK".

Sadly, there has been little communication between Everyone/Freely and the technical press. Freely employs a publicity agency, which describes itself as specialising in entertainment, arts and culture. Freely's own press staff appear to have little interest in, or understanding of, the technology that makes Freely tick. Some of the companies selling Freely hardware are equally difficult to talk to.

Viewers are used to near-instant switching between live broadcast channels. However, some delay is inevitable when streaming live TV because the incoming IP stream must be buffered and processed. Freely combines both broadcast and IP delivery, so delays may be uncomfortably obvious.

The amount of memory and processing power inside Freely hardware is inevitably fixed at the time of manufacture. The software and firmware can be updated, but not the electronics. This is why smart TVs and HDMI bolt-on streaming devices, like Roku and Amazon Fire sticks, become obsolete. The saving grace is that these sticks have generally been relatively cheap.

So far, at least, Freely bolt-on devices are expensive. The first to appear, Netgem's Pleio 'puck', costs around £100, largely because it has gaming features not everyone needs.

The 'cheaper' Aero from Manhattan costs around £70. The Humax AuraEZ 4K Recorder with "Built-in Freely" costs £250 (**https://humaxdirect.co.uk/products/humax-auraez-4k-freely-recorder**). Humax's advertising says:

*Just plug in and Aura EZ is ready to watch in minutes. Record up to four channels at a time while watching a fifth live, when connected to an aerial..Generous 2TB hard drive stores up to 1,000 hours of recordings.*

*The Humax Aura EZ app will make it simple for Aura EZ owners to navigate, plan and schedule recordings remotely from anywhere, so the shows and movies they love most will always be available to watch – even on the go...no aerial or dish required.*

However, the 'small print' warns: "The recording feature on Aura EZ requires an aerial.....Recording is not compatible with the Freely platform."

Humax has a tech-savvy spokeswoman who explains: "The recording function is via the aerial and the Freely part is a separate interface. They do overlap in that you can be in the Freely interface and press the record button, which then brings up a guide overlaid on the top, so it's integrated in that way. But you cannot record any Freely channels".

Although Freely's website is frustratingly fluffy, it promises clearly: "All you need is Wi-Fi".

Will customers read the Humax small print carefully? Or will they buy a Humax-with-Freely recorder, try unsuccessfully to record a stream, and return the box for a refund? The telltale will be whether Amazon sells the Aura but adds the standard customer warning about "high return rates".

Manhattan, a company which also has a tech-savvy spokesperson, explains: "There is no aerial input.. (on the Aura)...our view is that when you're buying a Freely box, you're specifically choosing streaming-first TV. Including an aerial socket would have...created a less optimal experience, because Freely TVs with both options prioritise the aerial signal – meaning you lose the ability to pause live TV..."

So far, the only radio stations on Freely are from the BBC. However, as Manhattan explains, "if you go to 'all apps' on the Manhattan Aero, 'Radioline' is included and it has a wide range of stations. It's a global network but knows you're in the UK and provides accordingly. There are also apps like Capital FM, Classic FM, Heart and others on the Manhattan Aero."

This is all starting to sound rather complicated...

The BBC uses a wide variety of data rates for different applications. The main BBC stations (inside the UK with HQ Geoblock, via internet

*continued on page 25...*

# Teach-In 2026

by Mike Tooley

## World of Wireless – An Introduction to Radio and Wireless Technology

## Part 12.7: Spectrum and vector network analysis

**In the last part, we explained** the principles and techniques that underpin a range of useful RF measurements, including voltage, power and frequency. Our *Hands-On* project featured an RF probe that can be used in conjunction with a conventional multimeter to make voltage measurements over the range of 100kHz to 100MHz.

This month, we focus on techniques for analysing RF networks and devices, including S-parameters, vector network analysis and spectrum analysis.

S-parameters provide a common framework for RF circuit design, allowing engineers to measure and optimise RF devices and systems. They streamline measurement processes, enhance modelling accuracy and help engineers develop designs that are stable and reliable. Whether it's adjusting antennas, creating filters or testing amplifiers, S-parameters are a fundamental RF engineering tool.

Vector network analysis (VNA) is an indispensable tool in RF engineering, enabling accurate measurement of network and device characteristics over a wide frequency range. It ensures optimal performance in the design and manufacture of antennas, filters, amplifiers, couplers and isolators.

Spectrum analysis (SA) is important when it's necessary to investigate the components present in an RF signal, enabling an assessment of spectral purity and spurious content.

### Measuring RF networks & devices

Using standard testing tools, measuring the effect of an RF signal on signals passing through a device can be quite challenging, often involving multiple checks of voltage and phase angle across different frequencies. The device you're examining might be as simple as a resistive attenuator or as complex as a broadband amplifier.

However, by considering the device as a 'black box'—focusing solely on input and output rather than its internal workings—you can simplify the task. This approach makes RF circuit design easier for engineers.

A simple network, such as an antenna or resistive load, may have only one port. With an antenna, RF power is received as an input and released as radiated energy. A load also receives RF power but transforms it into heat.

Amplifiers, attenuators and filters are all examples of devices with two ports. In such cases, one port (usually Port 1) is designated the input and connected to a signal source. The output port (usually Port 2) is designated the output and connected to a load (or subsequent device).

Since what happens between Port 1 and Port 2 can be very important, we need a way to describe and categorise it. There are several ways of doing this, but the S-parameters that we shall meet a little later provide us with arguably the most powerful and flexible method.

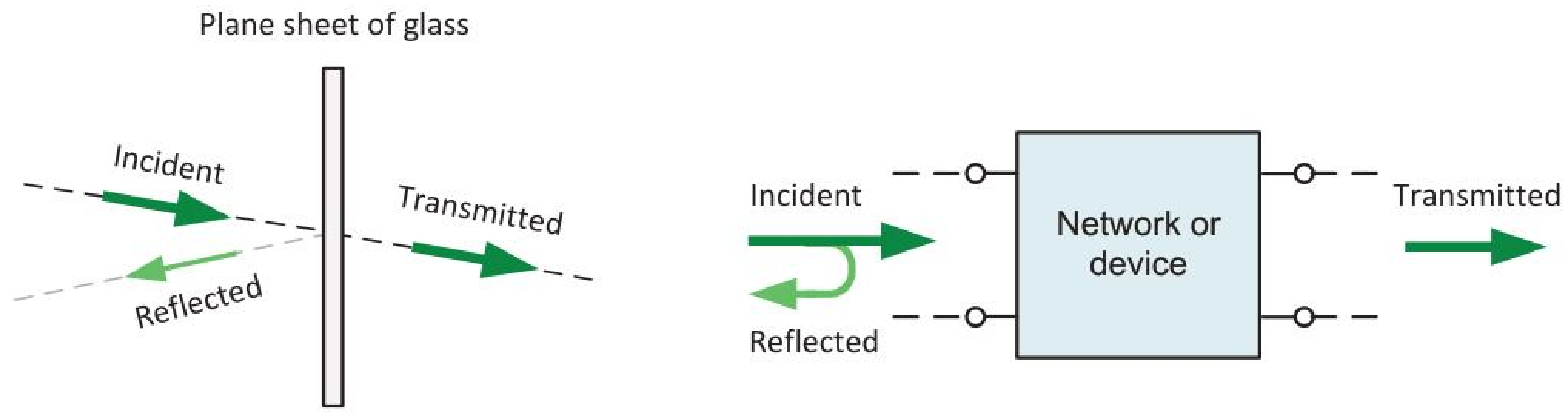
Some basic two-port networks based on passive devices are symmetrical, which means that either port can be used as the input or output without changing how the network functions and what it does. Common examples of this are matched resistive attenuators and passive filters.

In contrast, other two-port networks are asymmetrical and require specific connections for their input and output; swapping these ports will impact their operation. Amplifiers and multipliers are examples of asymmetric two-port networks. Devices like combiners, circulators and certain filters need a third port.

As the number of ports in a network increases, more parameters are necessary to fully describe its characteristics.

### S-parameters

S-parameters provide an efficient means of characterising the behaviour of both single- and multi-port networks. The required number of S-parameters is determined by the total number of ports present: a single-port network can be described using just two parameters, a two-port network requires four and a three-port network is represented by eight parameters and so on.



(a) Light analogy

(b) RF energy

Fig.7.1: like light, RF energy can be reflected.

S-parameters are complex vector quantities that have both real and imaginary components. Since complex numbers are similar to vectors, they can instead be considered as a magnitude and an angle or phase.

S-parameters are named because of the ‘scattering’ effect of energy entering and leaving a network or device under test (DUT). Fig.7.1 illustrates this effect and shows how it’s analogous to light (which is also a form of electromagnetic energy) passing through a planar sheet of glass.

Reflection and scattering of incident light occurs on one side of the glass, while a small proportion of the transmitted light is reflected light from objects on the other side of the glass, passing a second time through it and emerging back towards the observer, as depicted in Fig.7.1(a).

Fig.7.1(b) shows a two-port RF network or DUT represented as a black box. Some energy is transmitted and appears at the output while, due to an imperfect match at the input, a proportion of the energy is reflected towards the source. The effect of reflected RF power was discussed earlier, in Part 5, which dealt with antennas, matching and voltage standing wave ratio (VSWR).

S-parameters are ratios and therefore have no units. It’s important to remember that they represent complex quantities, meaning they have both magnitude and phase. If you’re unfamiliar with this, it can seem confusing at first.

Think of it like a boat crossing a lake while a steady crosswind is blowing. The boat’s motion relative to the water and the wind’s effect are two vectors acting at an angle to each other. The actual path of the boat is determined by the vector sum of these effects.

Even though you may only see the boat moving across the water, its final trajectory depends on both components. You cannot predict where it will end up by considering only one.

In electronics, AC currents and voltages behave the same way. Their magnitudes alone are not sufficient; phase angle matters. The total current at a node is found by vector addition, not simple arithmetic addition, because each current has both magnitude and direction (phase).

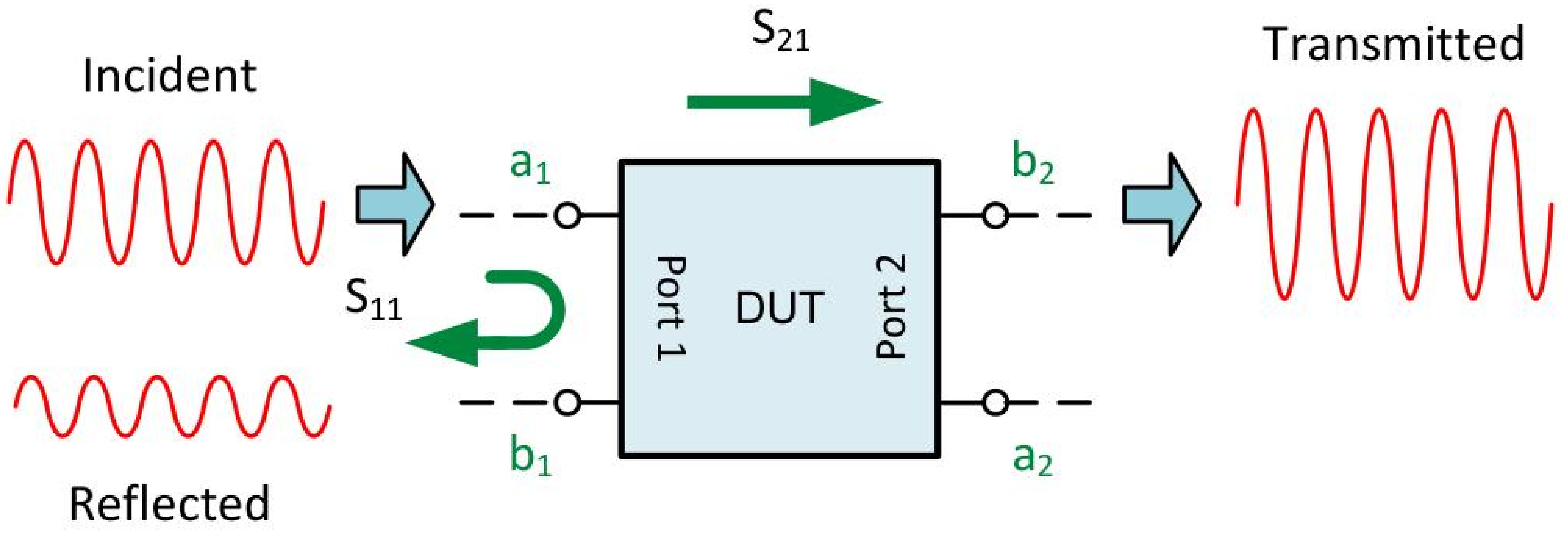
### Incident and reflected waves

S-parameters are best understood when conceptualised as dynamic wave quantities rather than static values. Waves enter and exit a network through its ports, necessitating a method to distinguish between them.

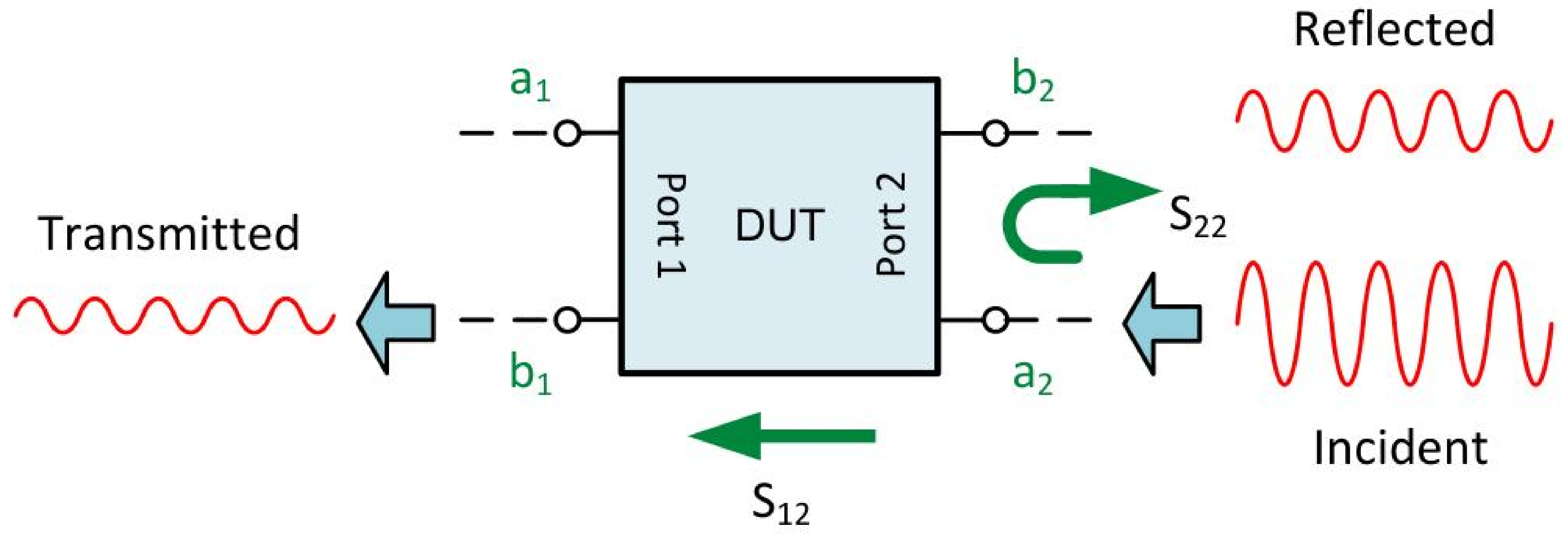
A vector network analyser (VNA) fulfils this role by separating incident waves from reflected waves, thereby enabling efficient and accurate characterisation of a network or device.

In Fig.7.2, waves entering and exiting the two-port network are labelled “a” and “b”, with subscripts 1 and 2 indicating the input (Port 1) and output (Port 2), respectively. At Port 1, the incoming wave is labelled a1 and the reflected wave is labelled b1. At Port 2, the transmitted (outgoing) wave is a2 and the reflected wave from the load is b2.

The ratios of these waves provide us with four S-parameters.



(a) Signal applied to Port 1 with matched load at Port 2



(b) Signal applied to Port 2 with matched load at Port 1

Fig.7.2: a generic two-port network showing incident, transmitted and reflected waves.

**Input Reflection Coefficient, S11**, where (with a2 = 0):

$$S_{11}=\frac{\text{Reflected at Port 1}}{\text{Forward at Port 1}}=\frac{b_1}{a_1}$$

S11 tells us how much of the signal is reflected at the input. This parameter is useful when evaluating impedance matching. With an ideal match at the input b1 is zero, therefore S11 is also zero.

**Forward Transmission (gain or loss), S21**, where (with a2 = 0):

$$S_{21}=\frac{\text{Transmitted from Port 1 to Port 2}}{\text{Incident at Port 1}}=\frac{b_2}{a_1}$$

S21 tells us how much gain or loss is present. This parameter is useful when determining voltage gain (in the case of an amplifier) or loss (in the case of an attenuator or passive filter).

**Output Reflection Coefficient, S22**, where (with a1 = 0):

$$S_{22}=\frac{\text{Reflected at Port 2}}{\text{Incident at Port 2}}=\frac{b_2}{a_2}$$

S22 tells us how much of the signal is reflected at the output. This parameter is useful when evaluating impedance matching. With an ideal match at the output b2 is zero, therefore S22 is also zero.

**Reverse Transmission, S12**, where (with a1 = 0):

$$S_{12}=\frac{\text{Transmitted from Port 2 to Port 1}}{\text{Incident at Port 2}}=\frac{b_1}{a_2}$$

S12 tells us how much signal from the output appears back at the input. This parameter is useful when analysing isolators, circulators and directional couplers.

### Vector network analysers

The measurement of S-parameters relies on matched impedances being present at all ports. Therefore, it is essential that both signal sources and loads are matched to a uniform impedance, typically 50Ω.

It is also necessary to ensure that the VNA is calibrated before use. This generally involves the use of known short-circuit and open-circuit loads.

Low-cost VNAs (some available for less than £100), like the one

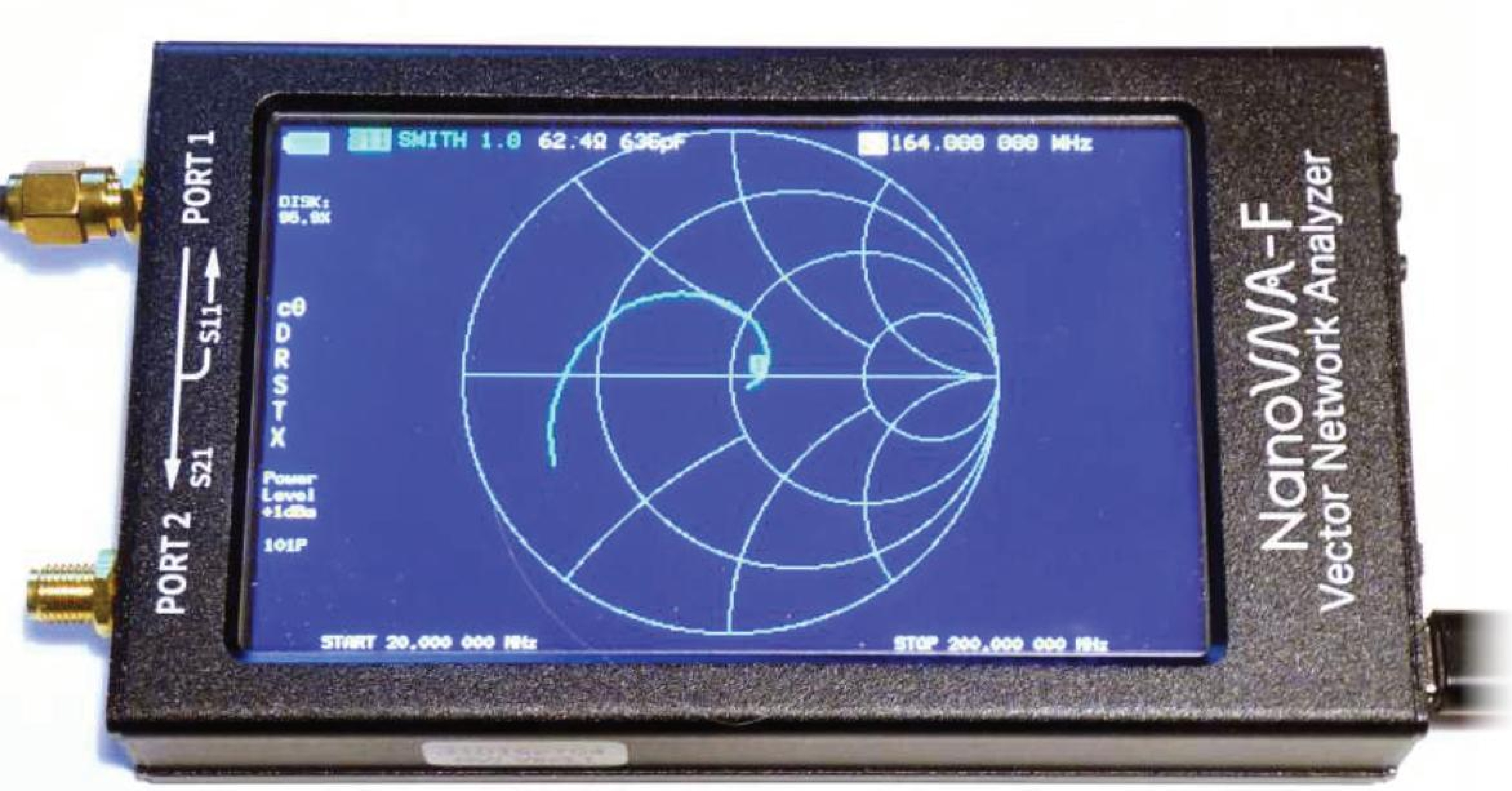


Fig.7.3: this inexpensive VNA can measure S11 and S21 from 50kHz to 1GHz.

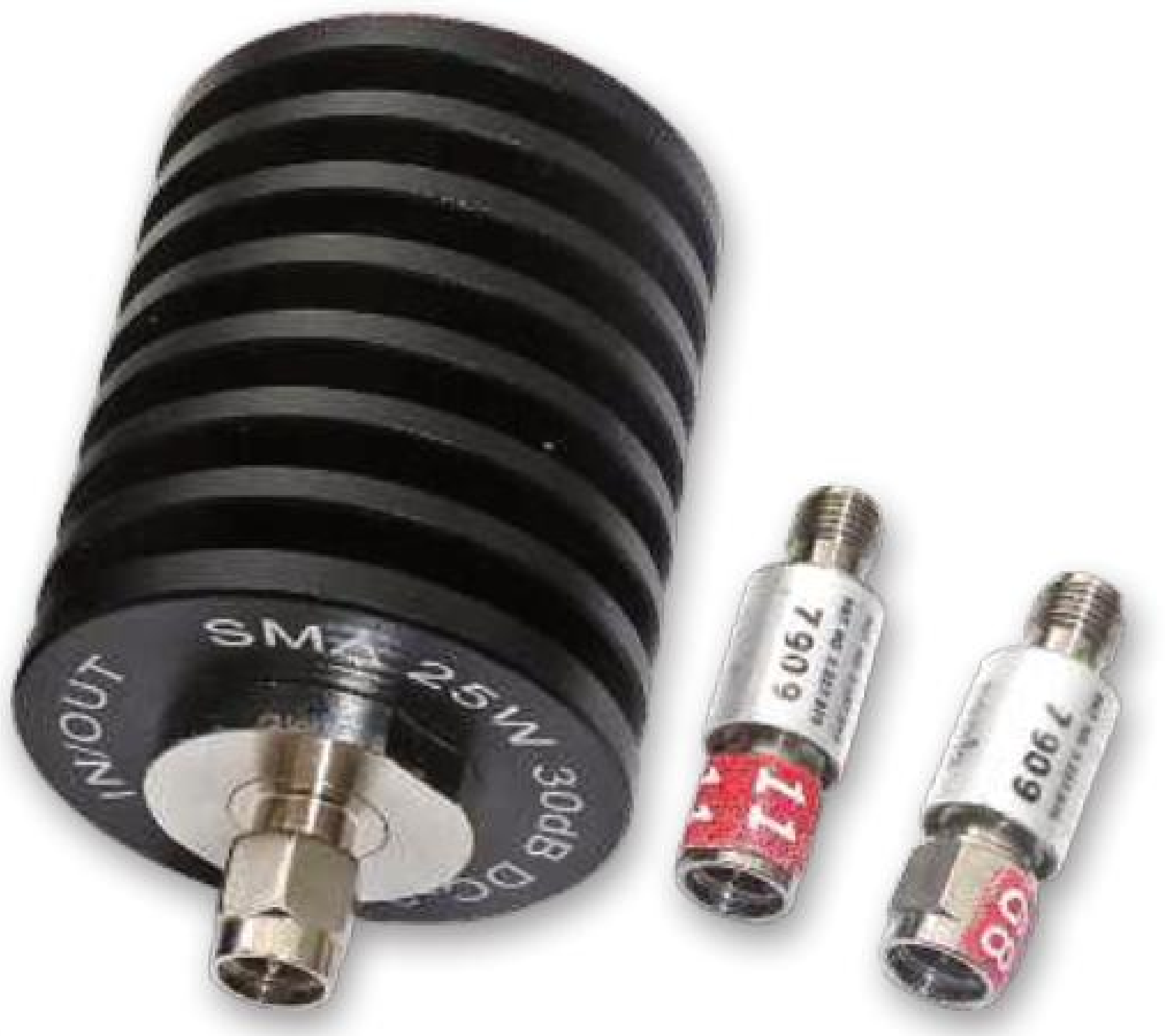
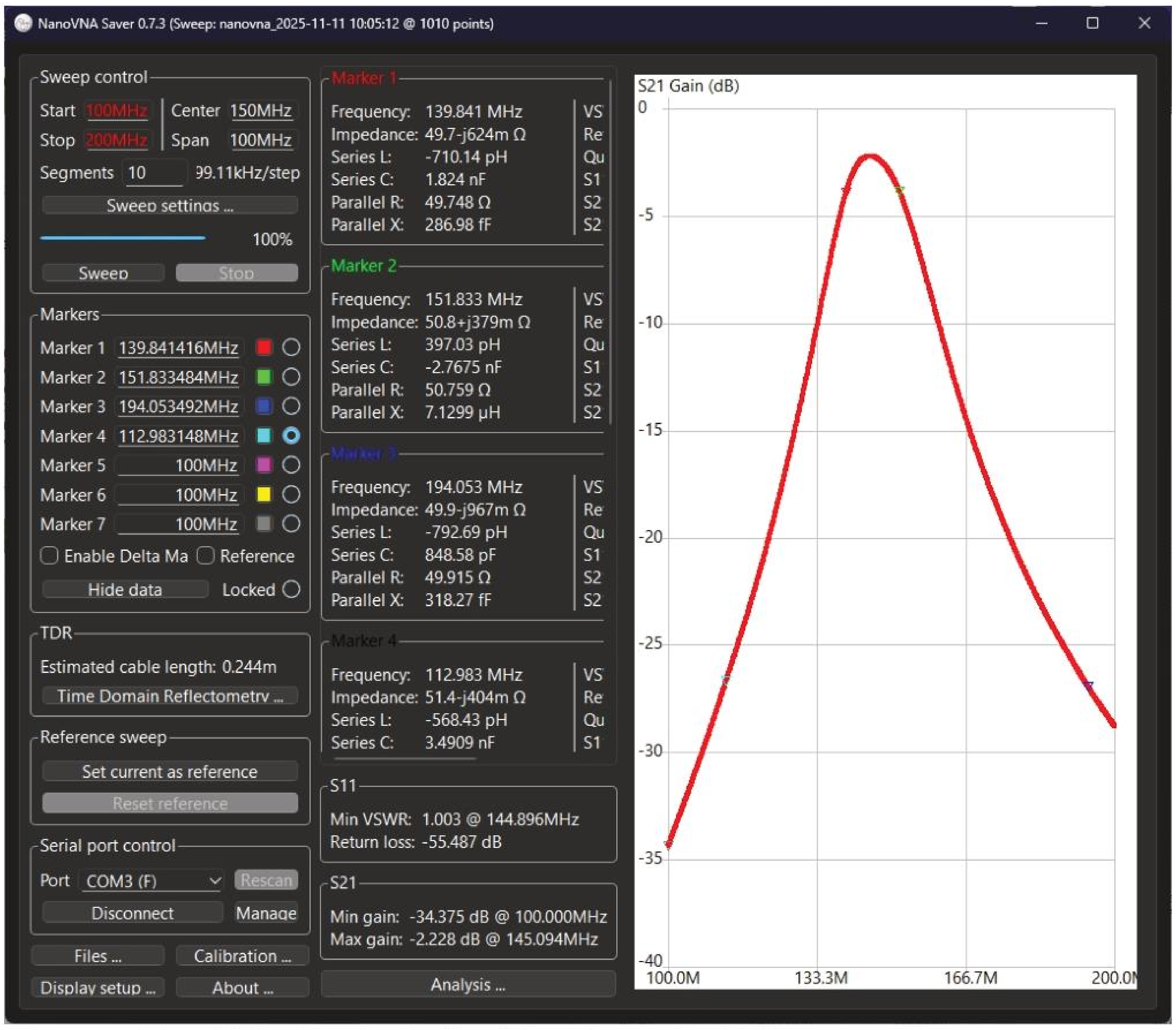


Fig.7.5: fixed external attenuators are often necessary to avoid overdriving a device and to minimise the risk of overloading the VNA's input.


NanoVNA Saver 0.7.3 (Sweep: nanovna_2025-11-11 10:05:12 @ 1010 points)

Sweep control

Start 100MHz
Stop 200MHz
Center 150MHz
Span 100MHz
Segments 10
99.11kHz/step
100%

Markers

Marker 1 139.841416MHz
Marker 2 151.833484MHz
Marker 3 194.053492MHz
Marker 4 112.983148MHz ●
Marker 5 100MHz
Marker 6 100MHz
Marker 7 100MHz
☐ Enable Delta Ma ☐ Reference
Locked ○

TDR

Estimated cable length: 0.244m

Serial port control

Port COM3 (F)

Marker 1

Frequency: 139.841 MHz
Impedance: 49.7-j624m Ω
Series L: -710.14 pH
Series C: 1.824 nF
Parallel R: 49.748 Ω
Parallel X: 286.98 fF

Marker 2

Frequency: 151.833 MHz
Impedance: 50.8+j379m Ω
Series L: 397.03 pH
Series C: -2.7675 nF
Parallel R: 50.759 Ω
Parallel X: 7.1299 μH

Marker 3

Frequency: 194.053 MHz
Impedance: 49.9-j967m Ω
Series L: -792.69 pH
Series C: 848.58 pF
Parallel R: 49.915 Ω
Parallel X: 318.27 fF

Marker 4

Frequency: 112.983 MHz
Impedance: 51.4-j404m Ω
Series L: -568.43 pH
Series C: 3.4909 nF

S11

Min VSWR: 1.003 @ 144.896MHz
Return loss: -55.487 dB

S21

Min gain: -34.375 dB @ 100.000MHz
Max gain: -2.228 dB @ 145.094MHz

Fig.7.4: a NanoVNA plot of gain (S21) for a low-noise VHF preamplifier.

shown in Fig.7.3, measure just S11 and S21. Despite that obvious limitation, these two parameters can still be extremely useful, making such a VNA a useful and affordable tool for those on a limited budget.

Laboratory-grade instruments can carry out measurements on multiple ports, but can be prohibitively expensive. For most amateur applications, a frequency range extending to 1GHz will be perfectly adequate, but it's worth checking that an instrument can take many samples.

A USB connection to a PC can be invaluable for analysing and displaying data using an external software package, such as the immensely popular NanoVNA Saver.

The S-parameter data captured by a VNA can be processed and displayed in various ways, including a list of parameters showing real and imaginary quantities and charts with data plotted against frequency.

**Table 7.1: partial S-parameter data for a dipole ant.**

| # HZ S RI R 50 | | |
|---|---|---|
| 135000000 | 0.179377839 | -0.305732042 |
| 135019822 | 0.176021755 | -0.306564718 |
| 135039644 | 0.172860488 | -0.307987154 |
| 135059466 | 0.169660151 | -0.309449017 |

Table 7.1 shows the first few (of several hundred) lines of data produced by a low-cost VNA when measuring the S-parameters of the dipole antenna featured in Teach-In Part 5. The data for this single port device is stored in an s1p 'snap' or 'Touchstone' file. The data for two-port devices require correspondingly larger s2p files.

The first line of data in Table 7.1 reveals the following:

- The first column shows the test frequencies specified in Hertz (Hz)
- The next two columns are real (R) and imaginary (I) quantities (the impedance and phase angle can easily be determined from these values)
- The data has been normalised to an impedance of 50Ω (more of this a little later)

Taking the first line of data in Table 7.1, we can determine the impedance of our dipole antenna at 135MHz by simply multiplying the real and imaginary quantities by the normalising impedance (50Ω).

Thus, at 135MHz, the impedance of our antenna is approximately (8.95 – j15.3)Ω. The negative sign indicates that the antenna is capacitive at 135MHz, which is expected, as this frequency is below the antenna's fundamental resonance.

Fig.7.4 shows a typical VNA application in which the forward gain characteristic (S21) of a low-noise VHF preamplifier is plotted from S-parameter data obtained using an inexpensive NanoVNA. To prevent the signal from the NanoVNA overloading the preamplifier, a 20dB attenuator has been inserted in the preamplifier input. This means that 20dB needs to be added to the gain visible in Fig.7.4.

The summary data for S21 thus shows that the preamplifier is delivering a gain of approximately 20dB – 2dB = 18dB.

Fig.7.5 shows some SMA attenuators that provide an attenuation of 30dB at up to 25W (left) and 10dB at up to 2W (right).

Although eminently suited for portable and handheld use, smaller

VNAs often have displays of limited size. To overcome this problem and provide more advanced data processing and display, VNA data can usually be exported in s1p or s2p format to a computer using an SD memory card or USB link.

Fig.7.6 shows RSplot being used to display s2p data obtained from a low-cost 6MHz filter. S11 (plotted in dB) is shown in red while S21 (also plotted in dB) is shown in blue. Markers at 6.4MHz have been superimposed on the characteristic.

Notice how S11 is relatively flat, while S21 clearly shows how the attenuation increases significantly at around 6.45MHz. Using this data, it's easily possible to determine filter bandwidth and shape factor.

AppCAD is another excellent (and free) program for manipulating and displaying data from S-parameters. Fig.7.7 shows how this extensive package can be used to test and confirm the response of an HF linear amplifier. Notice how S21 has been selected in the "Parameters to Display" window and an offset of 30dB has been chosen in the "Plot Options".

Adjustments for the external attenuator (Fig.7.5) can then be made automatically. It's evident from the frequency response that the quoted specifications have been exceeded, with the gain remaining reasonably constant over the HF range from 5MHz to 30MHz.

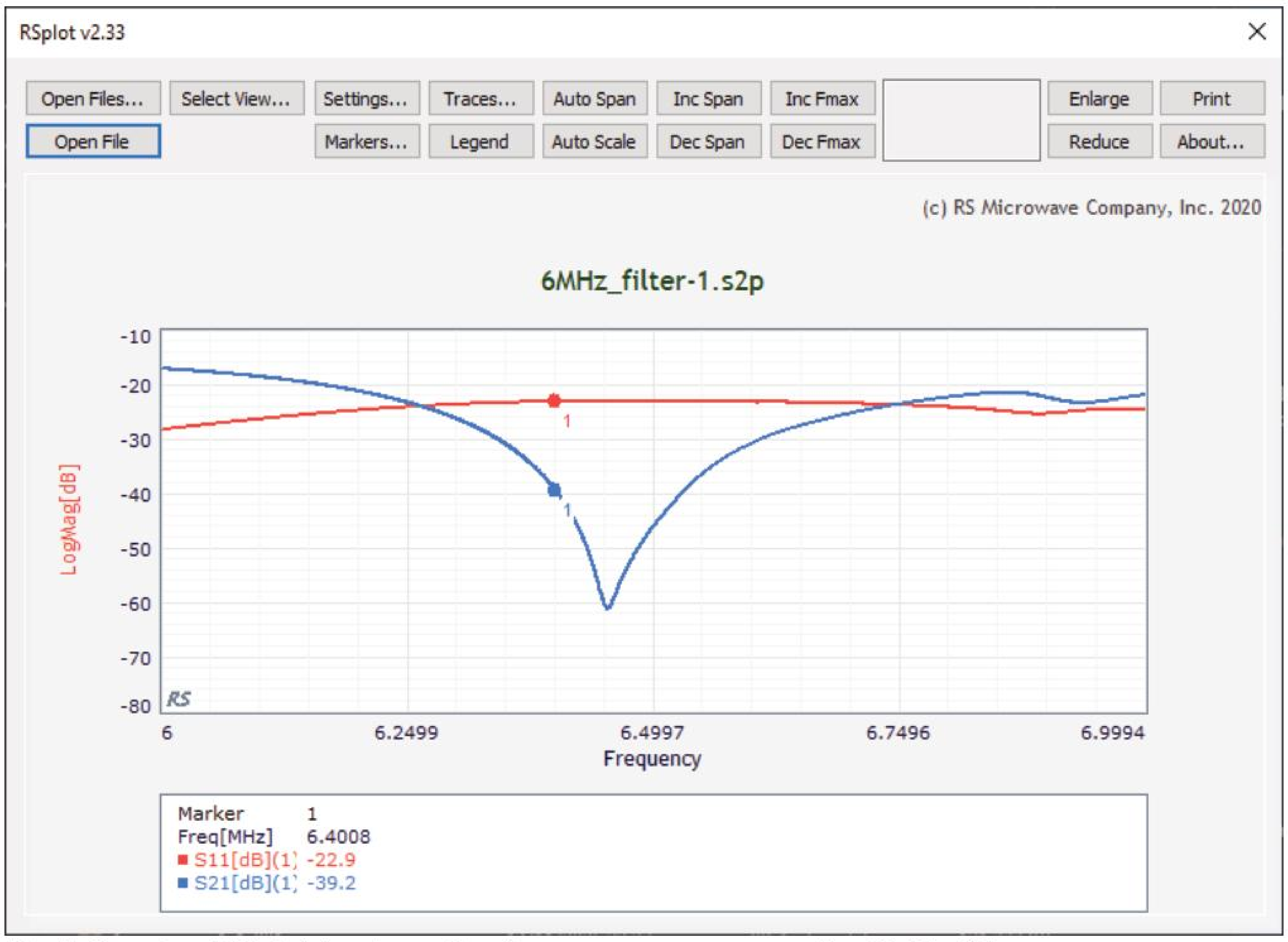
## The Smith chart

To get the best out of a VNA, it's important to be familiar with a method of graphical representation known as a Smith chart. We will start by explaining how the Smith chart works before moving on to showing how a VNA can be used to perform some basic tasks.

A typical application is in the design of antennas and the arrangements for feeding and matching them to a transmitter. Fig.7.8 shows the basic arrangement of a Smith chart.

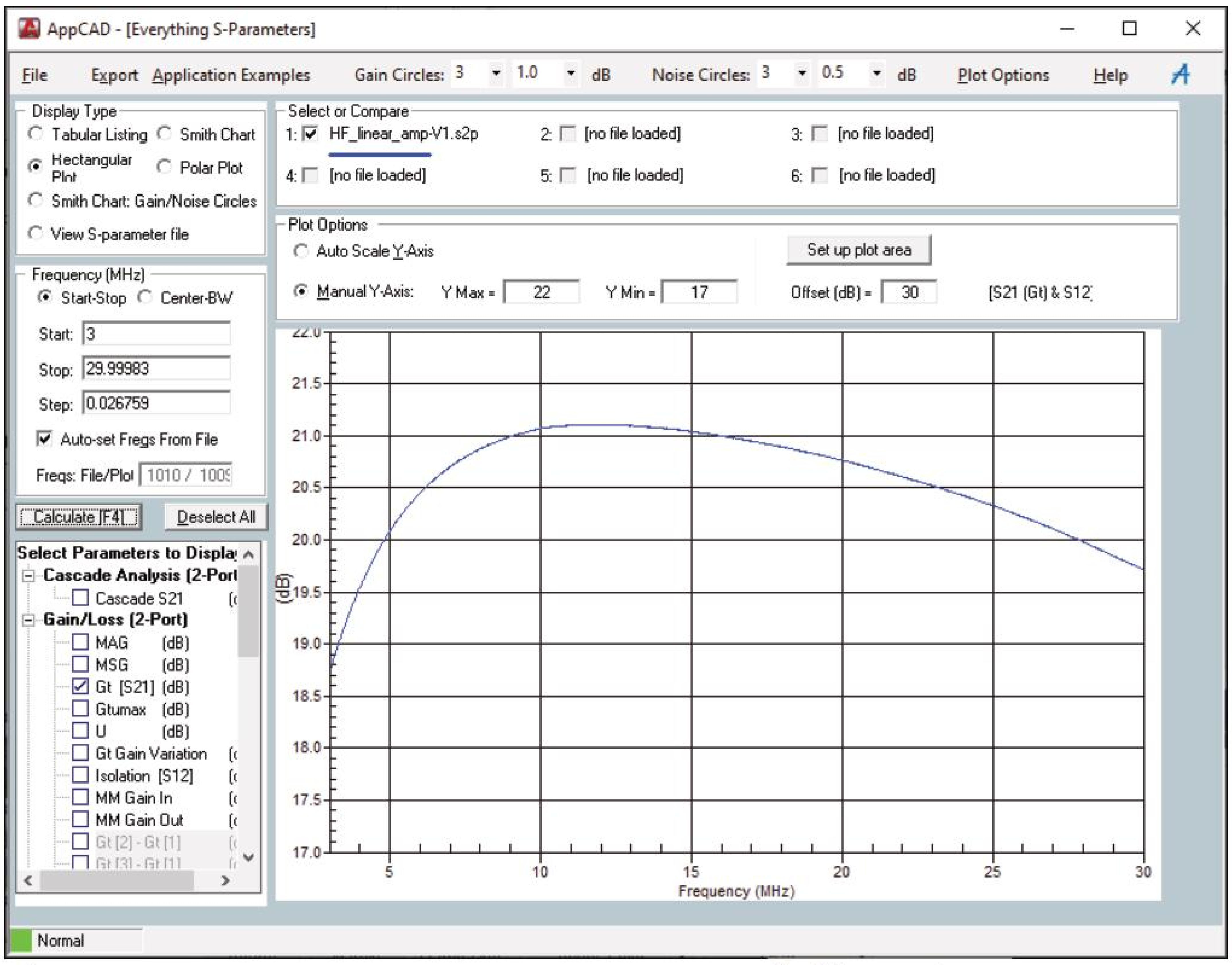
Although this may look somewhat daunting, the basic concept is simply that a complex impedance (an impedance comprising both real/resistive and imaginary/reactive components) can be mapped into a circular space in which the horizontal axis represents resistance (the real part) and the outer circumference represents reactance (the imaginary part).

Positive values of reactance (inductive reactance) appear along the upper semi-circumference, while negative values of reactance (capacitive reactance) correspond to the lower semi-circumference. Imaginary (reactive) components are preceded by the letter j (instead of the i used by mathematicians), the sign of which can be either positive or negative according to whether the reactance is inductive or capacitive.


RSplot v2.33
Open Files...
Select View...
Settings...
Traces...
Auto Span
Inc Span
Inc Fmax
Enlarge
Print
Open File
Markers...
Legend
Auto Scale
Dec Span
Dec Fmax
Reduce
About...
(c) RS Microwave Company, Inc. 2020
6MHz_filter-1.s2p
LogMag[dB]
-10
-20
-30
-40
-50
-60
-70
-80
RS
1
1
6
6.2499
6.4997
6.7496
6.9994
Frequency

Marker 1
Freq[MHz] 6.4008
■ S11[dB](1) -22.9
■ S21[dB](1) -39.2

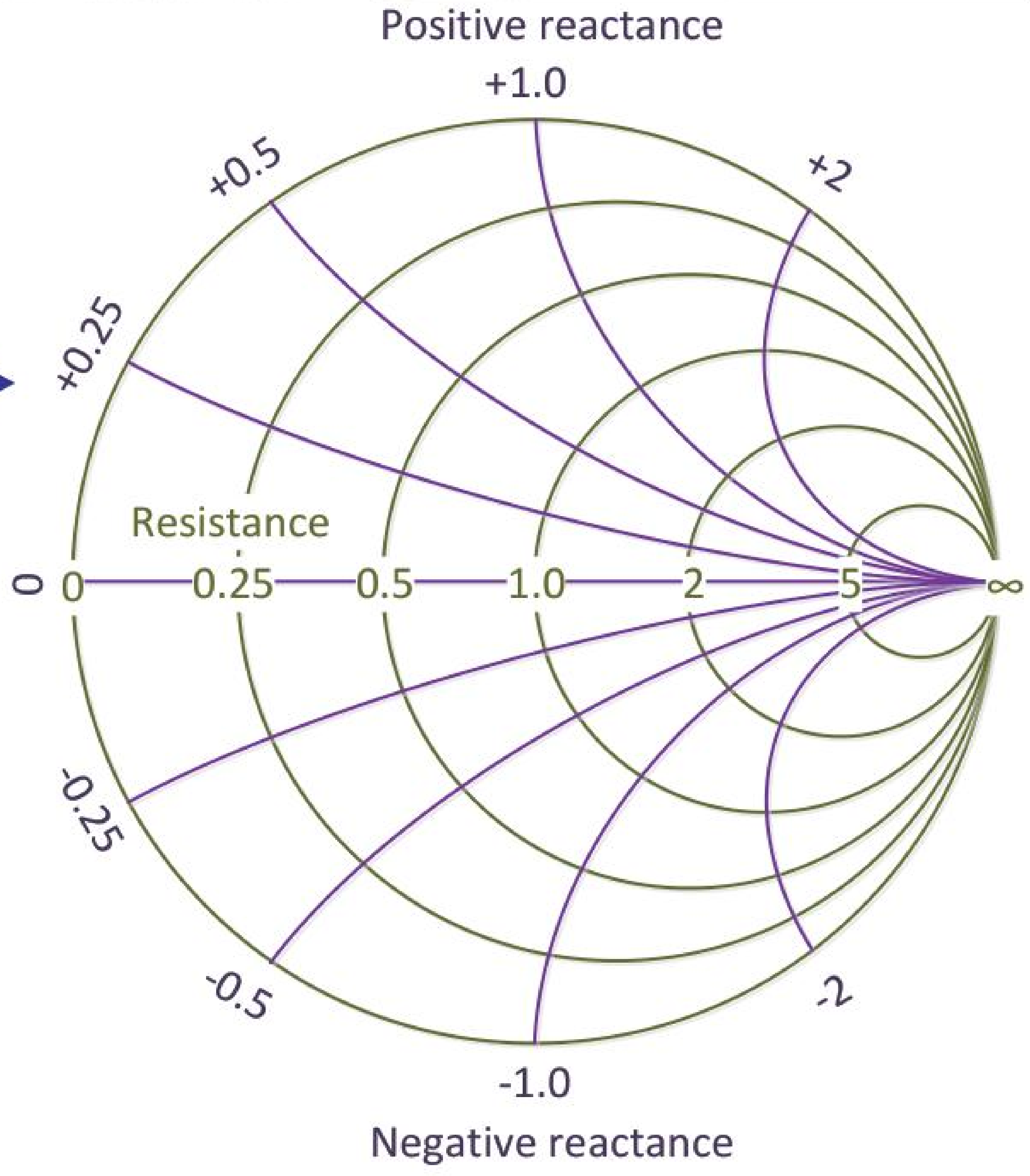
Fig.7.6: using RSplot to show the frequency response of a 6MHz filter.


AppCAD - [Everything S-Parameters]
File
Export
Application Examples
Plot Options
Help
Display Type
○ Tabular Listing
○ Smith Chart
● Rectangular Plot
○ Polar Plot
○ Smith Chart: Gain/Noise Circles
○ View S-parameter file
Select or Compare
Plot Options
○ Auto Scale Y-Axis
Set up plot area
Frequency (MHz)
○ Center-BW
Calculate [F4]
Deselect All
Select Parameters to Display
Cascade Analysis (2-Port)
☐ Cascade S21
Gain/Loss (2-Port)
☐ MAG (dB)
☐ MSG (dB)
☑ Gt [S21] (dB)
☐ Gtumax (dB)
☐ U (dB)
☐ Gt Gain Variation
☐ Isolation [S12]
☐ MM Gain In
☐ MM Gain Out
☐ Gt [2] - Gt [1]
☐ Gt [3] - Gt [1]
(dB)
22.0
21.5
21.0
20.5
20.0
19.5
19.0
18.5
18.0
17.5
17.0
5
10
15
20
25
30
Frequency (MHz)
Normal

Gain Circles: 3 1.0 dB
Noise Circles: 3 0.5 dB
1: ☑ HF_linear_amp-V1.s2p
2: ☐ [no file loaded]
3: ☐ [no file loaded]
4: ☐ [no file loaded]
5: ☐ [no file loaded]
6: ☐ [no file loaded]
● Manual Y-Axis: Y Max = 22 Y Min = 17
Offset (dB) = 30 [S21 (Gt) & S12]
● Start-Stop
Start: 3
Stop: 29.99983
Step: 0.026759
☑ Auto-set Freqs From File
Freqs: File/Plot 1010 / 1009

▲ Fig.7.7: the powerful AppCAD package measuring the gain and frequency response of an HF linear amplifier. It has a quoted gain of 20dB from 5MHz to 30MHz.

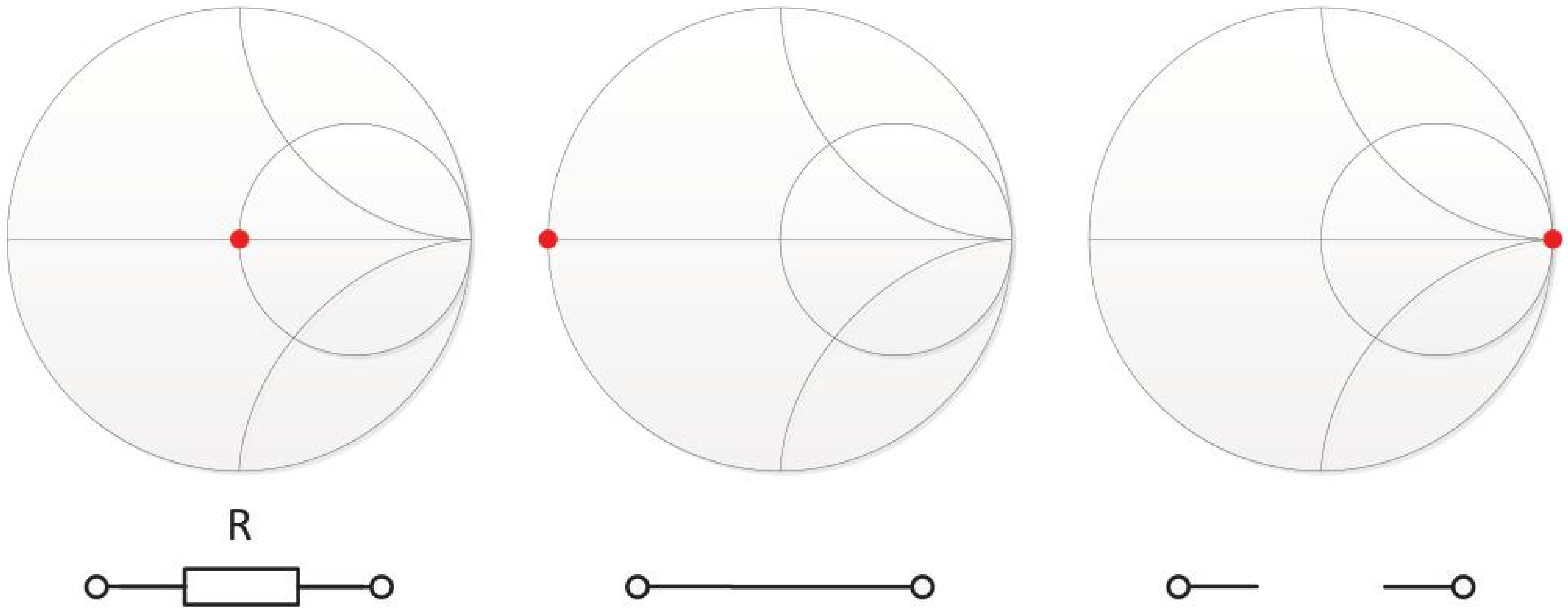
Fig.7.8: the basic arrangement of a Smith chart. ▶


Positive reactance
+1.0
+0.5
+2
+0.25
Resistance
0
0
0.25
0.5
1.0
2
5
∞
-0.25
-0.5
-2
-1.0
Negative reactance

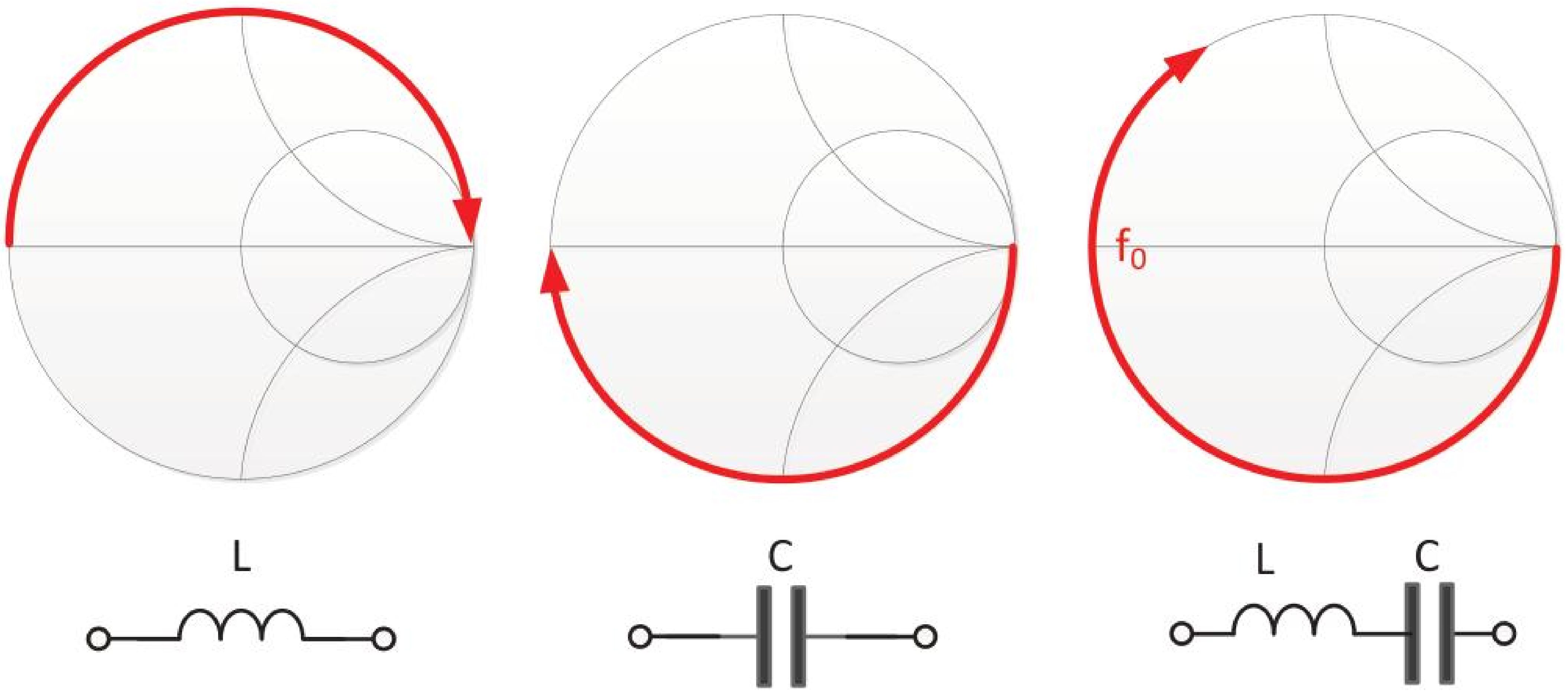
Some fundamental components are shown together with their Smith chart representation in Fig.7.9. A pure resistance identical to the system's characteristic impedance is shown in Fig.7.9(a), while short and open circuit connections are respectively shown in Fig.7.9(b) and 7.9(c).

Pure inductance and pure capacitance are represented by points on the circumference, as shown in Fig.7.9(d) and Fig.7.9(e). Because their reactance changes with frequency, they are shown as lines rather than dots, with the arrow on the line showing the direction of increasing frequency.

Finally, Fig.7.9(f) shows a series combination of pure inductance and pure capacitance. Note that the impedance of this combination will become zero at the frequency of resonance, $f_0$. This is the resonant frequency at which the inductive and capacitive reactance will become equal but of opposite direction and cancel each other out.



(a) Pure resistance, R = $Z_0$ (b) Short circuit, R = 0 (c) Open circuit, R = ∞



(d) Pure inductance, L (e) Pure capacitance, C (f) Series L and C

Fig.7.9: fundamental RF components plotted on Smith charts.

**j-notation**

If you're unfamiliar with j-notation, just remember that j is used to indicate the reactive part of an impedance (j = $\sqrt{-1}$). If you're not too happy with the notion of 'imaginary' numbers, just remember that when the sign of the j-term is positive, a component will be inductive and, conversely, when the sign of the j-term is negative, a component will be capacitive.

All practical RF components and antennas exhibit complex impedances, with both resistance and reactance present, and thus can be represented using j-notation. Importantly, complex impedance will vary with frequency due to the changing values of reactance present.

Consider the case of a component that can be represented by a resistance of 50Ω connected in series with an inductance of 100nH. At 100MHz, the reactance of the inductor will be approximately 63Ω, and the component will have an impedance that can be expressed as (50 + j63)Ω. At 140MHz, the component's inductive reactance will increase to 88Ω, so the impedance will then be (50 + j88)Ω.

Corresponding impedance values for a resistance of 50Ω connected in series with a capacitor of 20pF will be (50 – j80)Ω at 100MHz and (50 – j57)Ω at 140MHz.

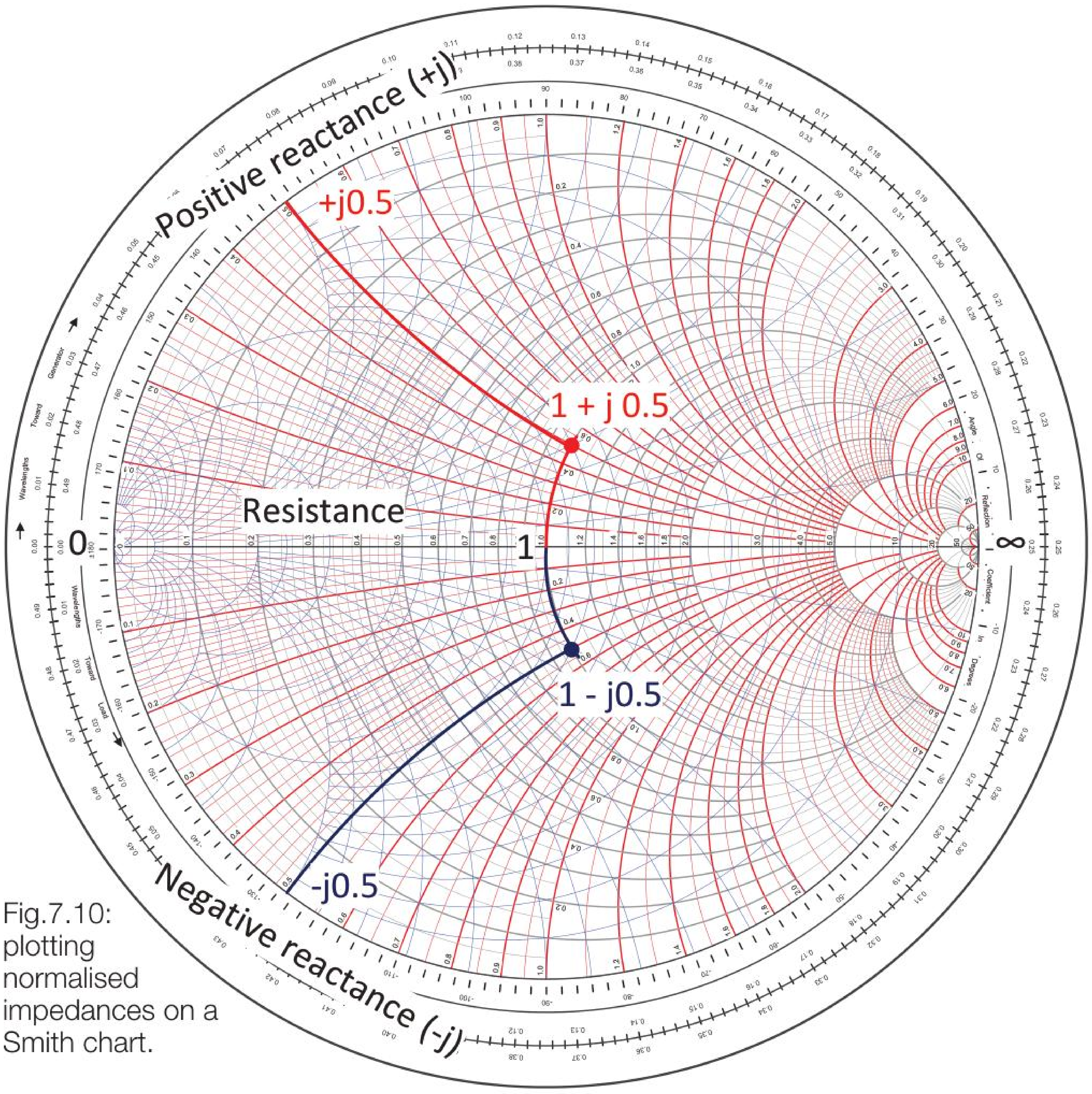


Fig.7.10: plotting normalised impedances on a Smith chart.

**Normalisation**

We briefly mentioned the need for normalising impedances in our VNA discussion. To cater for any given value of characteristic impedance, the values plotted on a Smith chart must first be normalised by simply dividing the real/resistive and imaginary/reactive parts by the value of characteristic impedance.

So, for example, an impedance of (50 + j25)Ω will be plotted as (1 + j0.5)Ω. Conversely, an impedance of (50 – j25)Ω will be normalised to (1 – j0.5)Ω. These conjugate points are shown plotted in Fig.7.10.

**Standing wave ratio**

The standing wave ratio can be

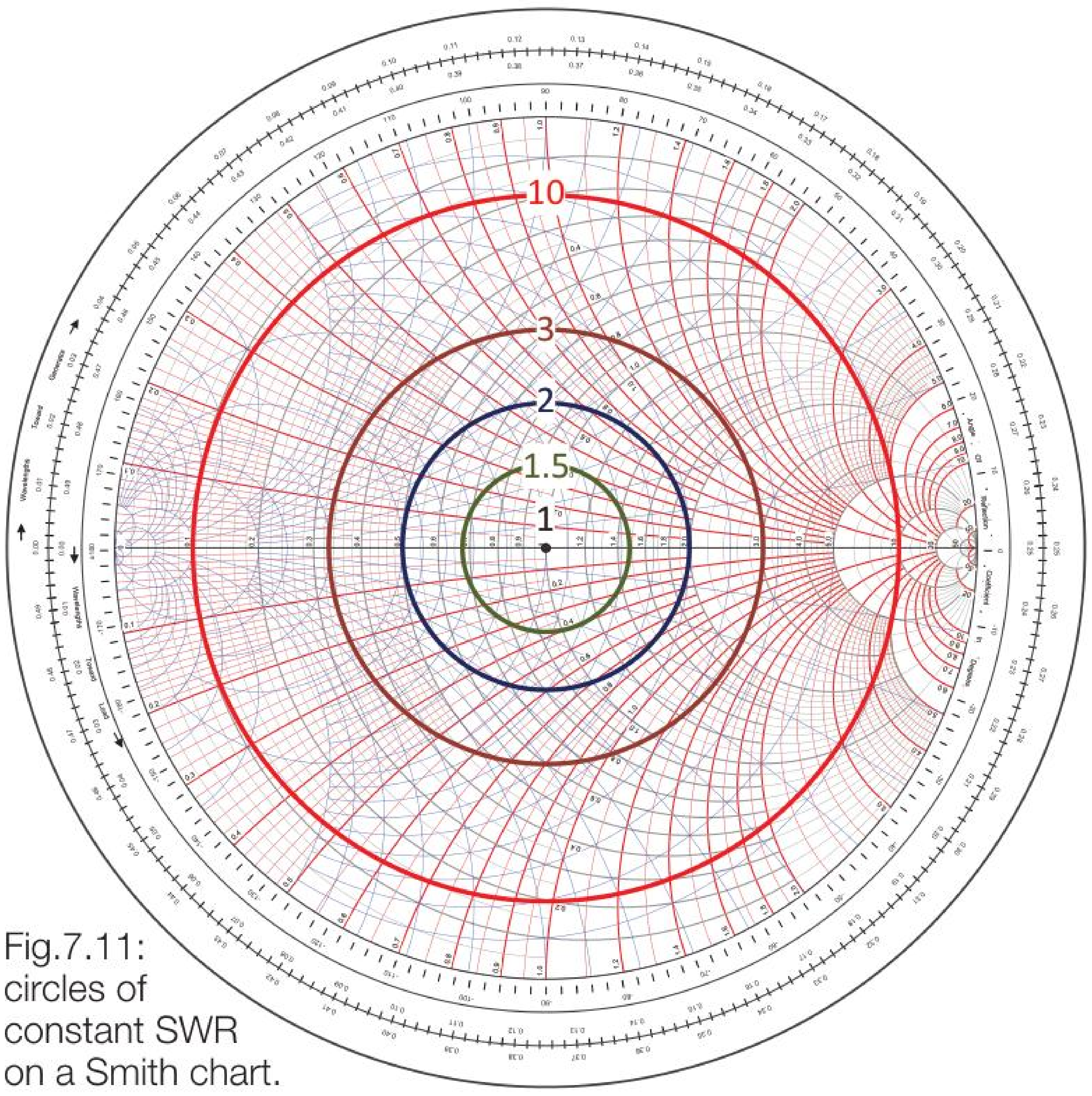


Fig.7.11: circles of constant SWR on a Smith chart.

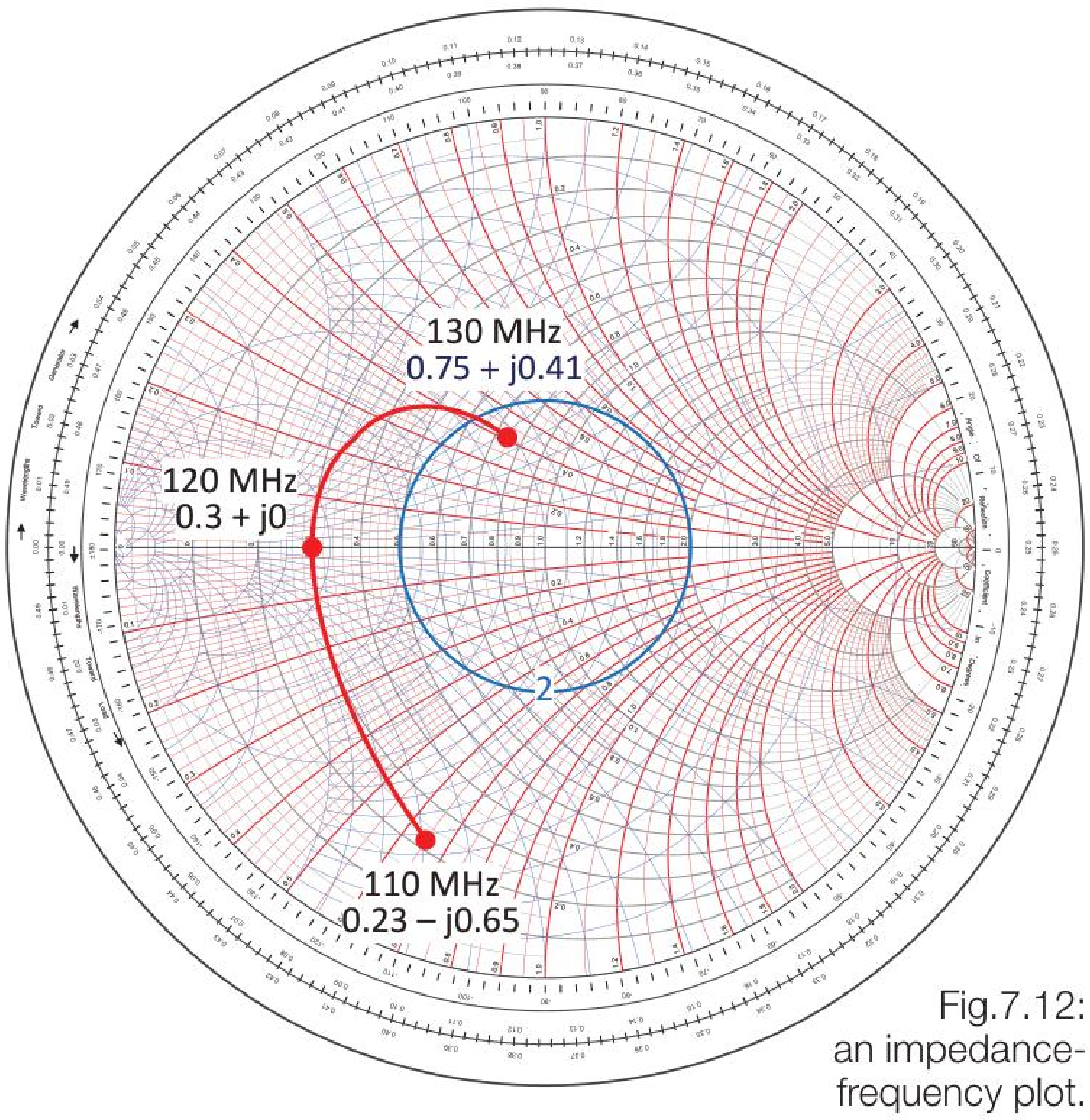
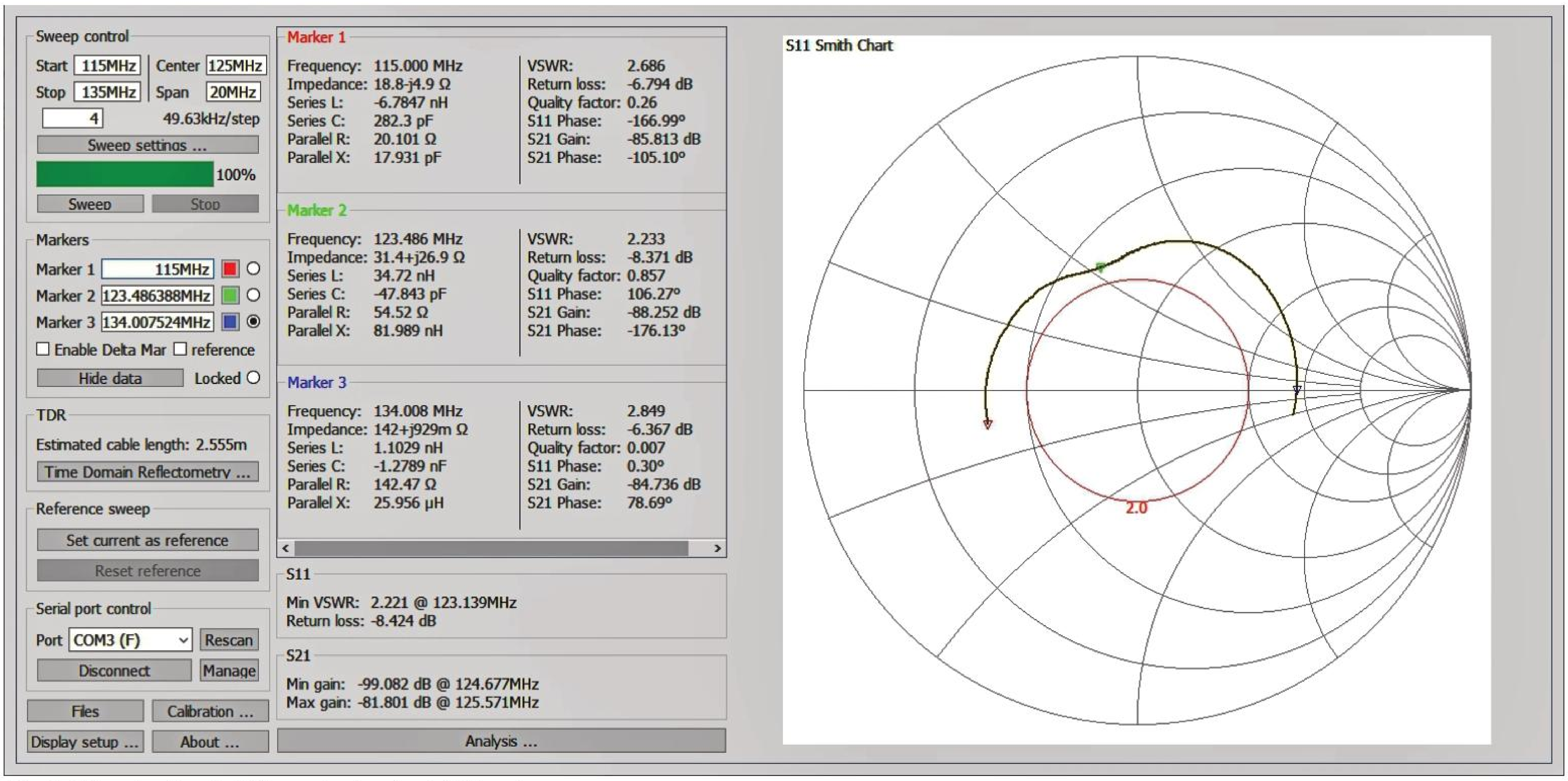


Fig.7.12: an impedance-frequency plot.



Sweep control

Start 115MHz | Center 125MHz

Stop 135MHz | Span 20MHz

4 49.63kHz/step

100%

Markers

Marker 1 115MHz ○

Marker 2 123.486388MHz ○

Marker 3 134.007524MHz ●

☐ Enable Delta Mar ☐ reference

Locked ○

TDR

Estimated cable length: 2.555m

Reference sweep

Serial port control

Port COM3 (F)

**Marker 1**

Frequency: 115.000 MHz
Impedance: 18.8-j4.9 Ω
Series L: -6.7847 nH
Series C: 282.3 pF
Parallel R: 20.101 Ω
Parallel X: 17.931 pF

VSWR: 2.686
Return loss: -6.794 dB
Quality factor: 0.26
S11 Phase: -166.99°
S21 Gain: -85.813 dB
S21 Phase: -105.10°

**Marker 2**

Frequency: 123.486 MHz
Impedance: 31.4+j26.9 Ω
Series L: 34.72 nH
Series C: -47.843 pF
Parallel R: 54.52 Ω
Parallel X: 81.989 nH

VSWR: 2.233
Return loss: -8.371 dB
Quality factor: 0.857
S11 Phase: 106.27°
S21 Gain: -88.252 dB
S21 Phase: -176.13°

**Marker 3**

Frequency: 134.008 MHz
Impedance: 142+j929m Ω
Series L: 1.1029 nH
Series C: -1.2789 nF
Parallel R: 142.47 Ω
Parallel X: 25.956 µH

VSWR: 2.849
Return loss: -6.367 dB
Quality factor: 0.007
S11 Phase: 0.30°
S21 Gain: -84.736 dB
S21 Phase: 78.69°

S11

Min VSWR: 2.221 @ 123.139MHz
Return loss: -8.424 dB

S21

Min gain: -99.082 dB @ 124.677MHz
Max gain: -81.801 dB @ 125.571MHz

Fig.7.13: a typical VNA analysis of a VHF antenna.

easily shown on a Smith chart by superimposing circles of constant SWR on the chart, as shown in Fig.7.11. Note that a unity value of SWR corresponds to a circle of zero radius (ie, a point at the exact centre of the Smith chart), while infinite SWR corresponds to the outer perimeter of the chart.

When used as a tool to investigate matching, the RF designer will invariably strive to ensure that load impedances are as close to the centre of the Smith chart as possible. Being able to see this all going on in real-time means making adjustments is a real breeze!

### Impedance-frequency plots

Smith charts are often used to visualise the behaviour of antennas and devices over a range of frequencies.

Fig.7.12 shows an example of the analysis of a component in which the normalised impedance varies from (0.23 – j0.65)Ω to (0.75 + j0.41)Ω over a frequency range extending from 110MHz to 135MHz.

Note how the antenna becomes purely resistive at 120MHz and how the SWR of the antenna has fallen below two above 130MHz.

### Antenna analysis

A low-cost vector network analyser (as in Fig.7.3) can provide a quick check on the characteristics of an antenna and its feeder. Fig.7.13 shows some typical results of the analysis of a 130MHz quarter-wave antenna in a 50Ω system. The analyser's frequency sweep has been set from 120MHz to 135MHz with markers displayed at 120MHz, 125MHz and 130MHz.

From Fig.7.13, we can quickly determine that:

- The antenna's SWR is less than 2 over the entire scan range
- The antenna is purely resistive at 123MHz and 133MHz
- The lowest SWR (approximately 1.6) is achieved at 130.5MHz; the best match to a 50Ω system is obtained at this frequency.

To help you get up to speed with a VNA, the low-cost RF Demo Kit

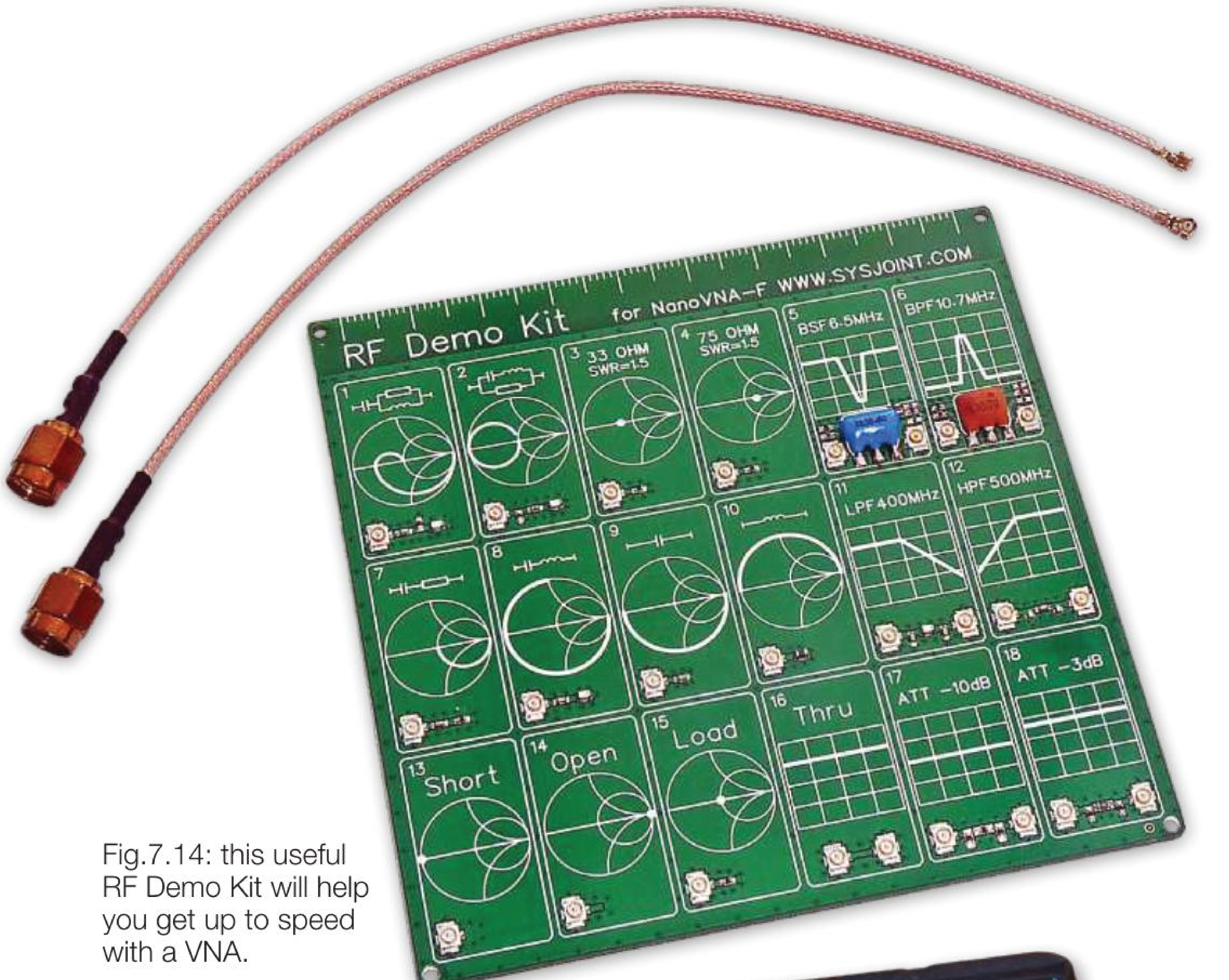

RF Demo Kit
for NanoVNA-F WWW.SYSJOINT.COM
33 OHM SWR=1.5
75 OHM SWR=1.5
BSF6.5MHz
BPF10.7MHz
LPF400MHz
HPF500MHz
Thru
ATT -10dB
ATT -3dB
Short
Open
Load

Fig.7.14: this useful RF Demo Kit will help you get up to speed with a VNA.

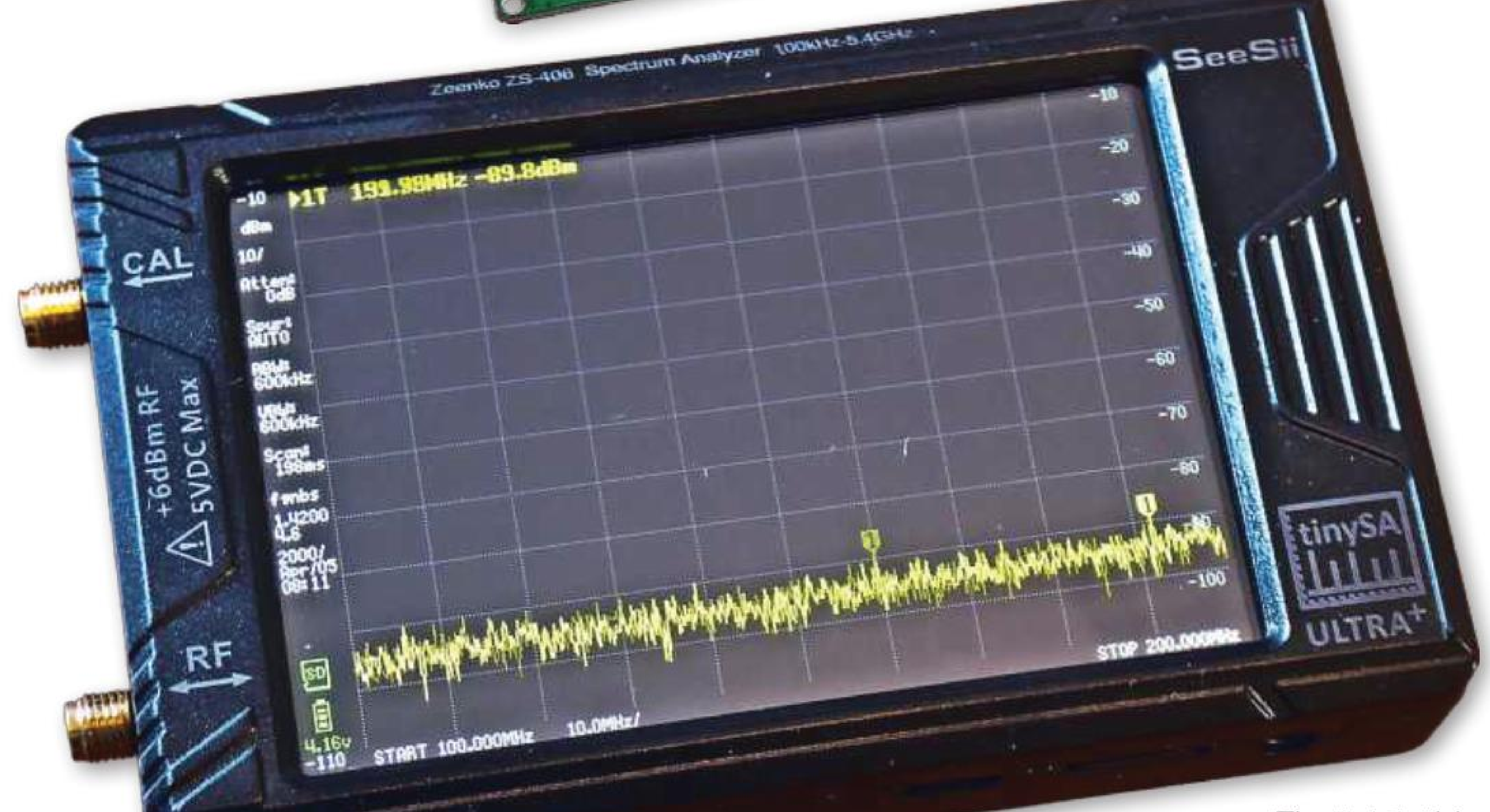

Zeenko ZS-406 Spectrum Analyzer 100kHz-5.4GHz
SeeSii
CAL
RF
+6dBm RF
5VDC Max
tinySA
ULTRA+
1T 199.98MHz -89.8dBm
START 100.000MHz
10.0MHz/
STOP 200.000MHz

Fig.7.15: this inexpensive handheld spectrum analyser operates from 100kHz to 5.4GHz

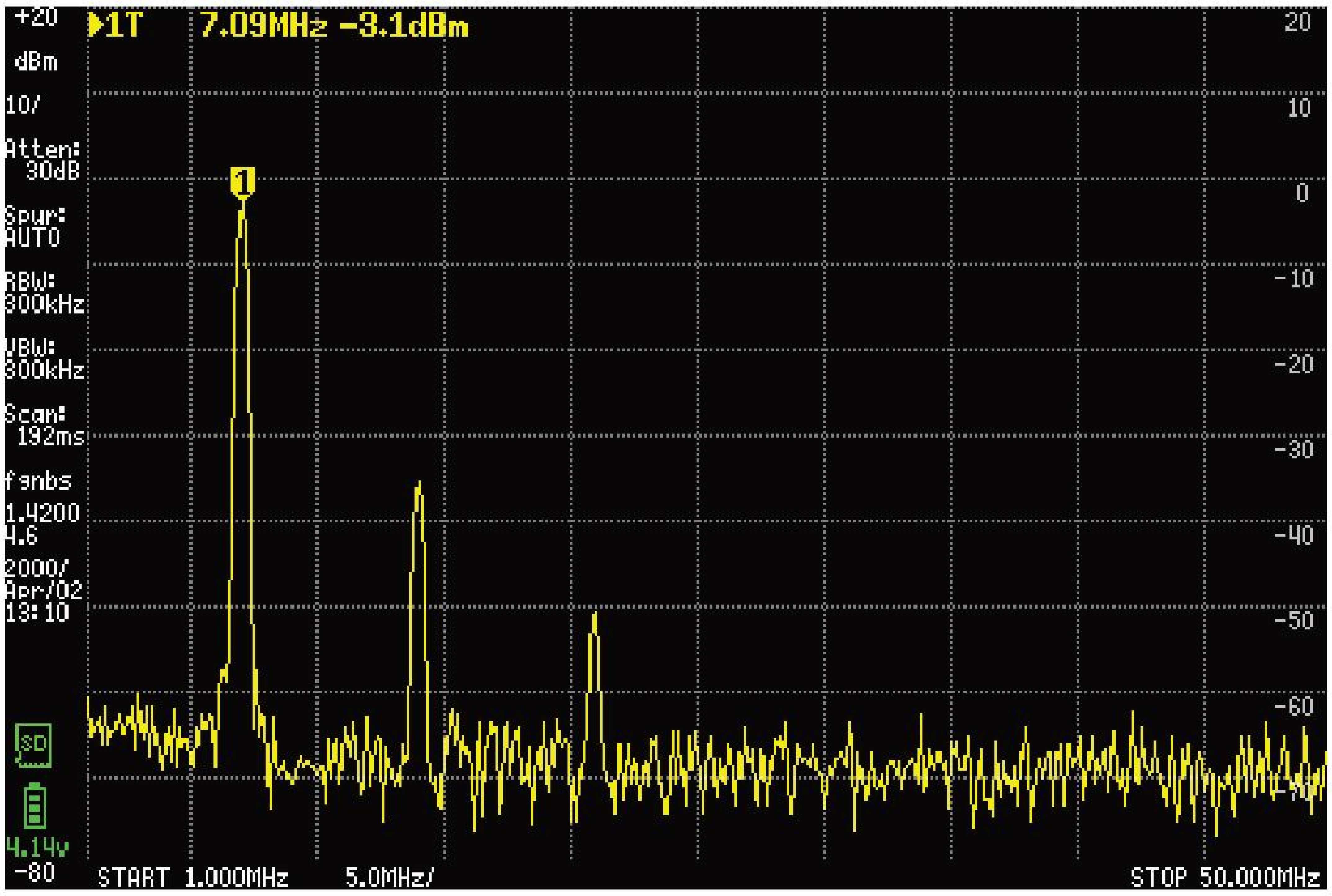

+20
dBm
10/
Atten:
30dB
Spur:
AUTO
RBW:
300kHz
VBW:
300kHz
Scan:
192ms
fanbs
1.4200
4.6
2000/
Apr/02
13:10
SD
4.14v
-80
1T 7.09MHz -3.1dBm
1
20
10
0
-10
-20
-30
-40
-50
-60
START 1.000MHz
5.0MHz/
STOP 50.000MHz

Fig.7.16: a typical screen grab from the spectrum analyser in Fig.7.15, showing the fundamental, 2nd and 3rd harmonic components in the output of a 7MHz transmitter.

shown in Fig.7.14 provides you with a variety of different devices for experimentation. This kit is available from several online suppliers and includes a handy set of PCB-mounted devices and circuits suitable for attachment to your VNA.

The Demo Kit includes short-circuit and open-circuit terminations as well as a low-power 50Ω load.

### Spectrum analysers

Spectrum analysers are invaluable tools for examining signals, enabling users to observe and interpret the output spectrum of RF sources including oscillators, synthesisers and transmitters.

By displaying the amplitude of signals across a wide range of frequencies, these instruments allow for the identification of fundamental signals, harmonic components, noise and any unwanted or spurious emissions.

Whether assessing the performance of transmitters, diagnosing interference, or ensuring compliance with regulatory standards, a spectrum analyser can provide you with a clear and immediate insight into the behaviour of a wide variety of RF systems. They are available in both benchtop and handheld versions, making them ideal for use both in the laboratory and in the field.

Laboratory-grade spectrum analysers can be prohibitively expensive, but in recent years, smaller portable instruments (as in Fig.7.15) have become increasingly affordable, yet still offer a reasonable level of performance.

For example, the tinySA analyser shown in Fig.7.15 provides a frequency coverage extending from 100kHz to 5.4GHz, making it suitable for examining a wide range of signals in the LF, MF, HF, VHF and UHF bands. Like the VNA that we met earlier, this handy device also supports a USB connection to a PC for charging and also for advanced data analysis and visualisation.

Even ‘proper’ benchtop instruments from brands like Rigol can be purchased starting at a few hundred pounds, although the cost goes up quickly if you start looking at the more capable instruments (eg, with a wider bandwidth).

Fig.7.16 shows a typical application for a low-cost spectrum analyser. This screengrab shows the output frequency spectrum of a home-built 7MHz transmitter. The frequency scan (horizontal axis) is set to cover the range from 1MHz to 50MHz with markers (dotted vertical lines) every 5MHz.

The fundamental (marked "1") is shown at 7.09MHz, at a measured level of -3.1dBm, corresponding to a power level of 0.49mW at the input of the spectrum analyser.

To prevent overloading the spectrum analyser's input and to provide a matched load for the transmitter, a matched 30dB attenuator was connected between the transmitter's output and the spectrum analyser input. Consequently, the true output power delivered by the transmitter will be 30dB greater than the power received by the spectrum analyser, ie, (30dBm – 3.1dBm) = 26.9dBm. This is equivalent to 0.49W from the transmitter.

As with a VNA, the ability to transfer data from the spectrum analyser to a PC opens further options for analysing and presenting data. Fig.7.17 shows an investigation of the output spectrum from a handheld digital VHF transceiver.

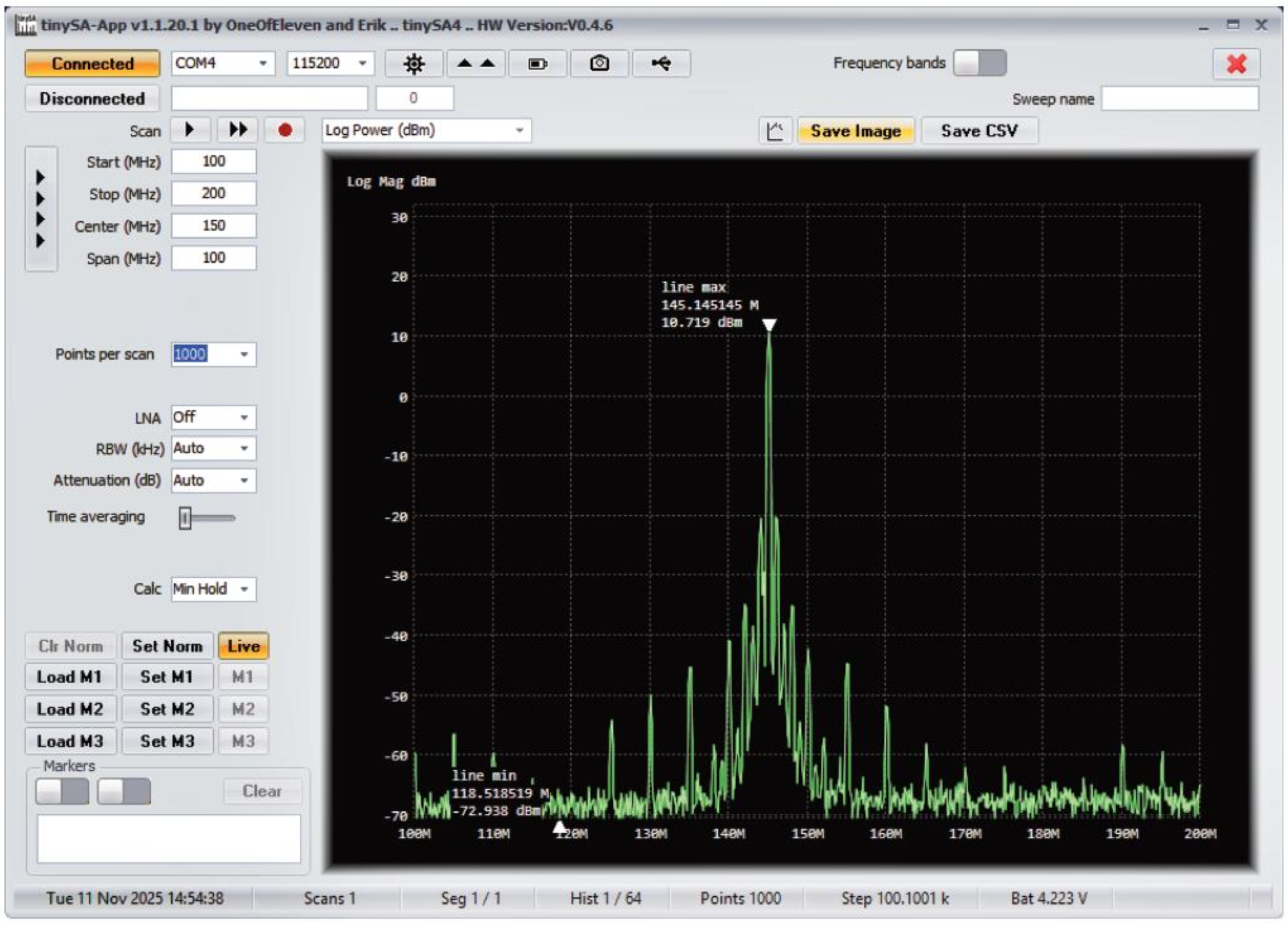
Note how the signal (at 145.145MHz) appears modulated by an unwanted component at 5MHz, resulting in an output spectrum that extends at 5MHz intervals for at least 50MHz on either side of the carrier!

The spectrum analyser's scan range can be adjusted to permit further investigation close to the carrier frequency where spurious content is at a level that exceeds the specified limits for the equipment under test. For example, setting the scan range from 130MHz to 160MHz will permit further examination of all spurious signals having a level of less than -60dB relative to the carrier.

Spectrum analysers can also provide a means of examining the RF spectrum over a very wide frequency range. This feature is invaluable when searching for sources of noise and interference that might cause difficulties with radio reception generally.

Fig.7.18 shows the spectrum received by a tinySA over the range 100kHz to 1GHz, enabling simultaneous display of the entire LF, MF, HF and VHF spectrum in real time.

In Fig.7.18, we've indicated the principal services present, as well as a strong locally generated carrier at 400MHz. Signal types can often be recognised by their shapes (due to the modulation method) viewed on the spectrum diagram. This can help you quickly distinguish AM, SSB, FM and digital signals.


tinySA-App v1.1.20.1 by OneOfEleven and Erik .. tinySA4 .. HW Version:V0.4.6

COM4
115200
0
Log Power (dBm)
Start (MHz) 100
Stop (MHz) 200
Center (MHz) 150
Span (MHz) 100
Points per scan 1000
LNA Off
RBW (kHz) Auto
Attenuation (dB) Auto
Calc Min Hold

line max
145.145145 M
10.719 dBm

line min
118.518519 M
-72.938 dBm

Tue 11 Nov 2025 14:54:38 | Scans 1 | Seg 1 / 1 | Hist 1 / 64 | Points 1000 | Step 100.1001 k | Bat 4.223 V

Fig.7.17: PC applications can further analyse spectrum data. Here, tinySA is showing unwanted (spurious) signals on the output of a handheld digital VHF transceiver.

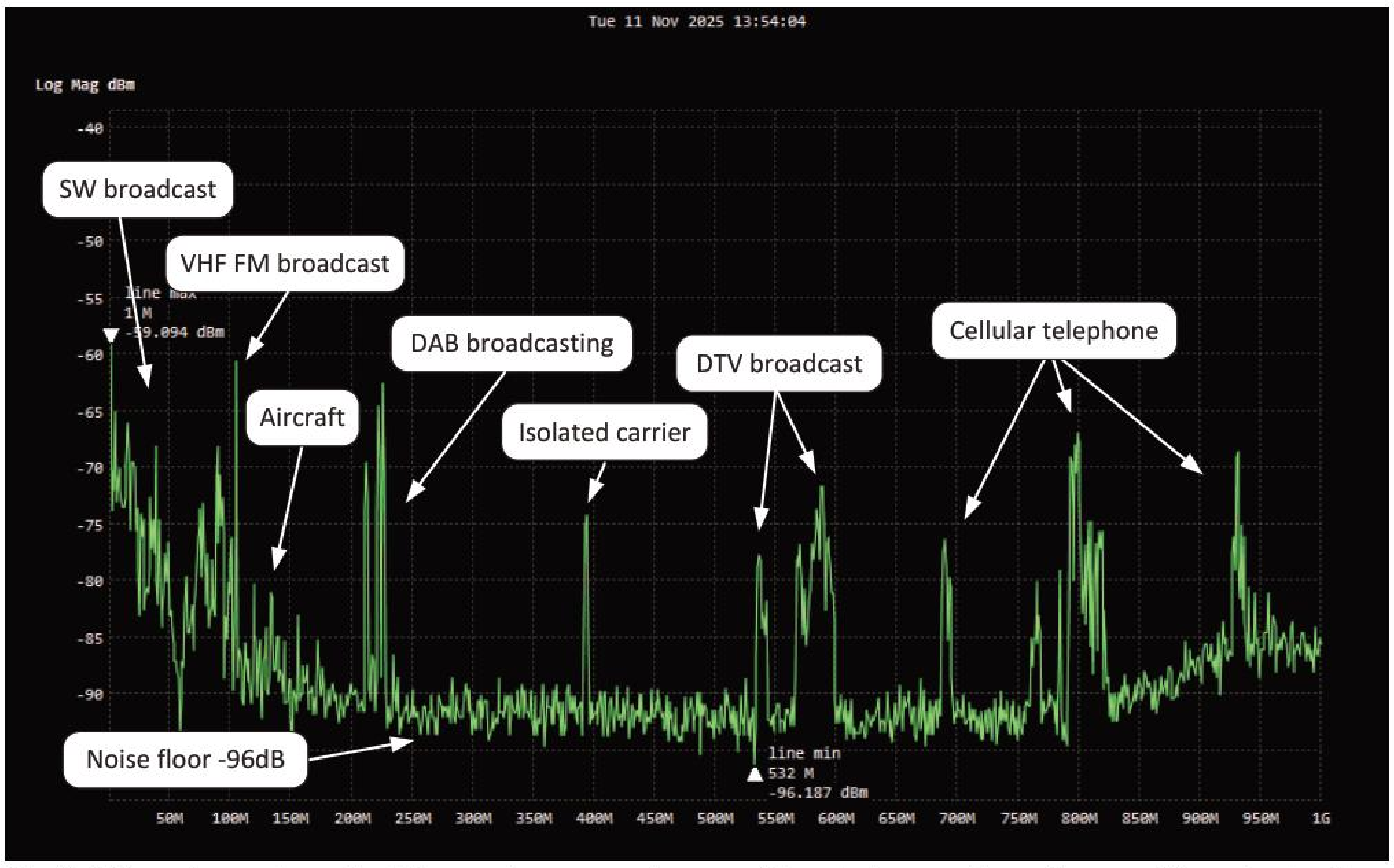
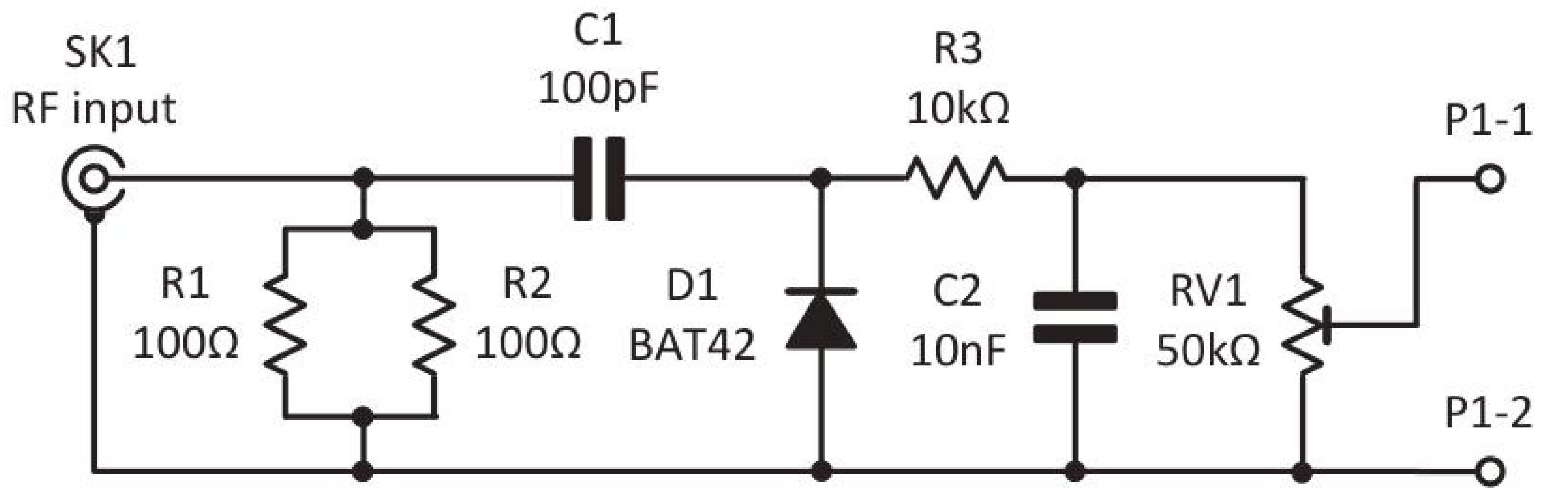


Fig.7.18: using a handheld spectrum analyser with a telescopic antenna. Thanks to its wide frequency coverage, we can view the LF, MF, HF and VHF spectra simultaneously.

Fig.7.19: the circuit of our 5W load and power meter.



### *Hands-On*: 5W load & power meter

This month's *Hands-On* project is a 5W load and power meter designed for measuring the output of low-power HF transmitters. The instrument is self-contained and its power indications are shown on an analog meter scale.

This makes it ideal for adjusting RF circuits and systems where it's easier to see the movement of a pointer over a calibrated scale than make sense of the fluctuating digits found on a conventional digital display. Although intended primarily for use with transmitters for the 80m (3.5MHz), 40m (7MHz) and 20m (14MHz) bands, the instrument can be used at frequencies of up to 30MHz.

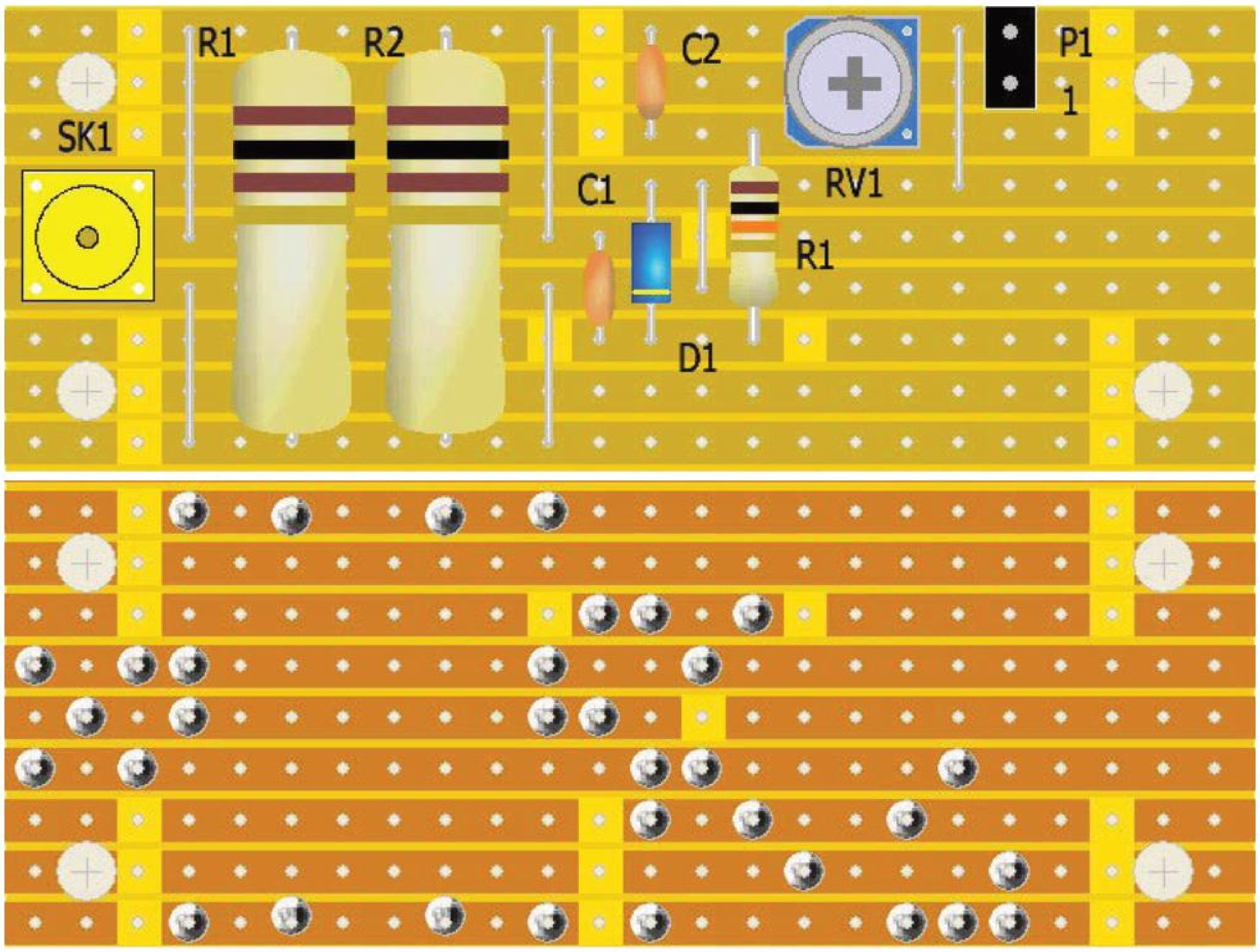
The load can dissipate 6W (see later) and this makes it ideal for use with low-power 'QRP' transmitters, which can normally deliver up to 5W.

The circuit of the load and power meter board is shown in Fig.7.19; the required 50Ω load resistance is obtained by connecting two 100Ω resistors (each rated at 3W) in parallel. The RF voltage developed across R1 and R2 is sensed and rectified by Schottky diode D1.

Resistor R3 and capacitor C2 form a low-pass filter, and the rectified DC output is developed across C2. Calibration adjustment is provided by RV1, and the output available at P1 is connected to a low-cost moving coil meter.

The component layout for the 5W load and power meter (viewed from the top) is shown in Fig.7.20 while the corresponding track layout (viewed from below) is given in Fig.7.21. The required 18 track breaks can be made using a spot face cutter or small drill bit, and the six links on the upper side of the boards are made using short lengths of tinned copper wire.

Note that R1 and R2 will run warm, so it's important to ensure that you leave space around (and under) these resistors to permit unrestricted airflow.



Figs.7.20 & 21: top and underside views of the 5W load and power meter board.

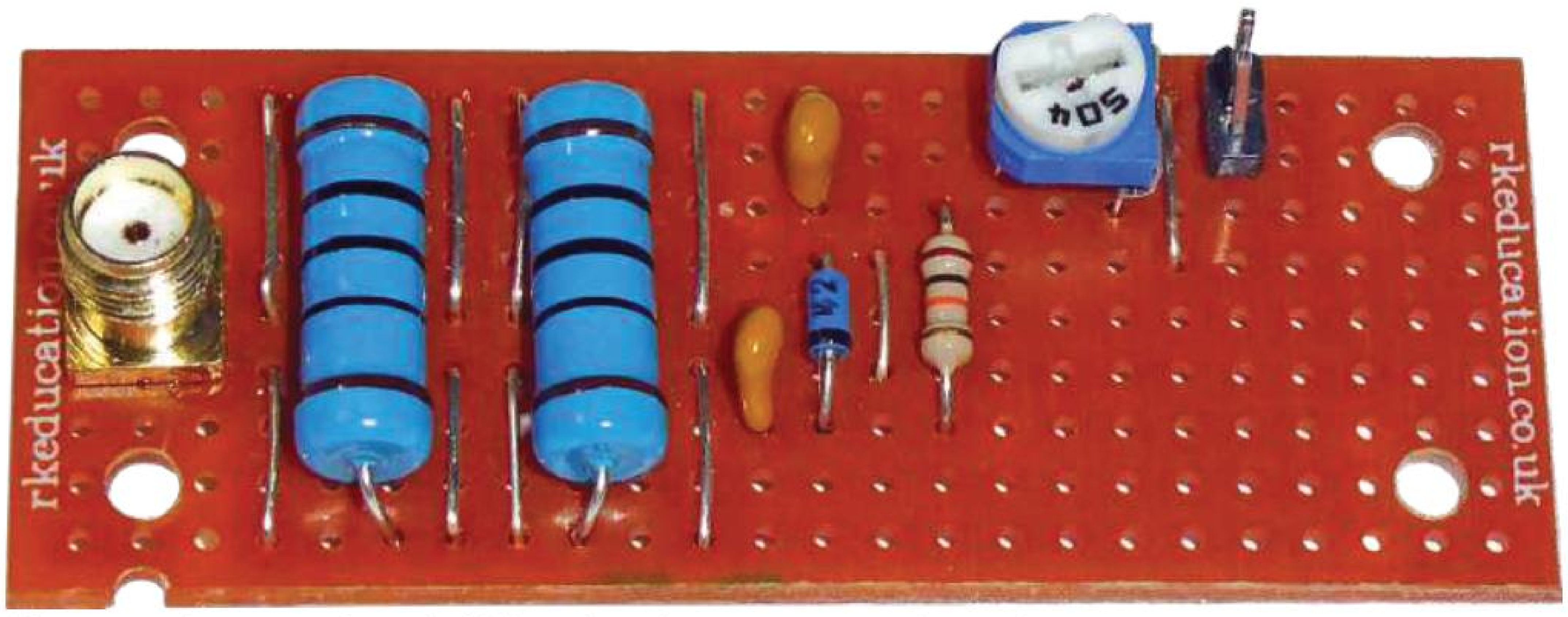


Fig.7.22: the completed 5W load and power meter board.

### Building the 5W load & power meter

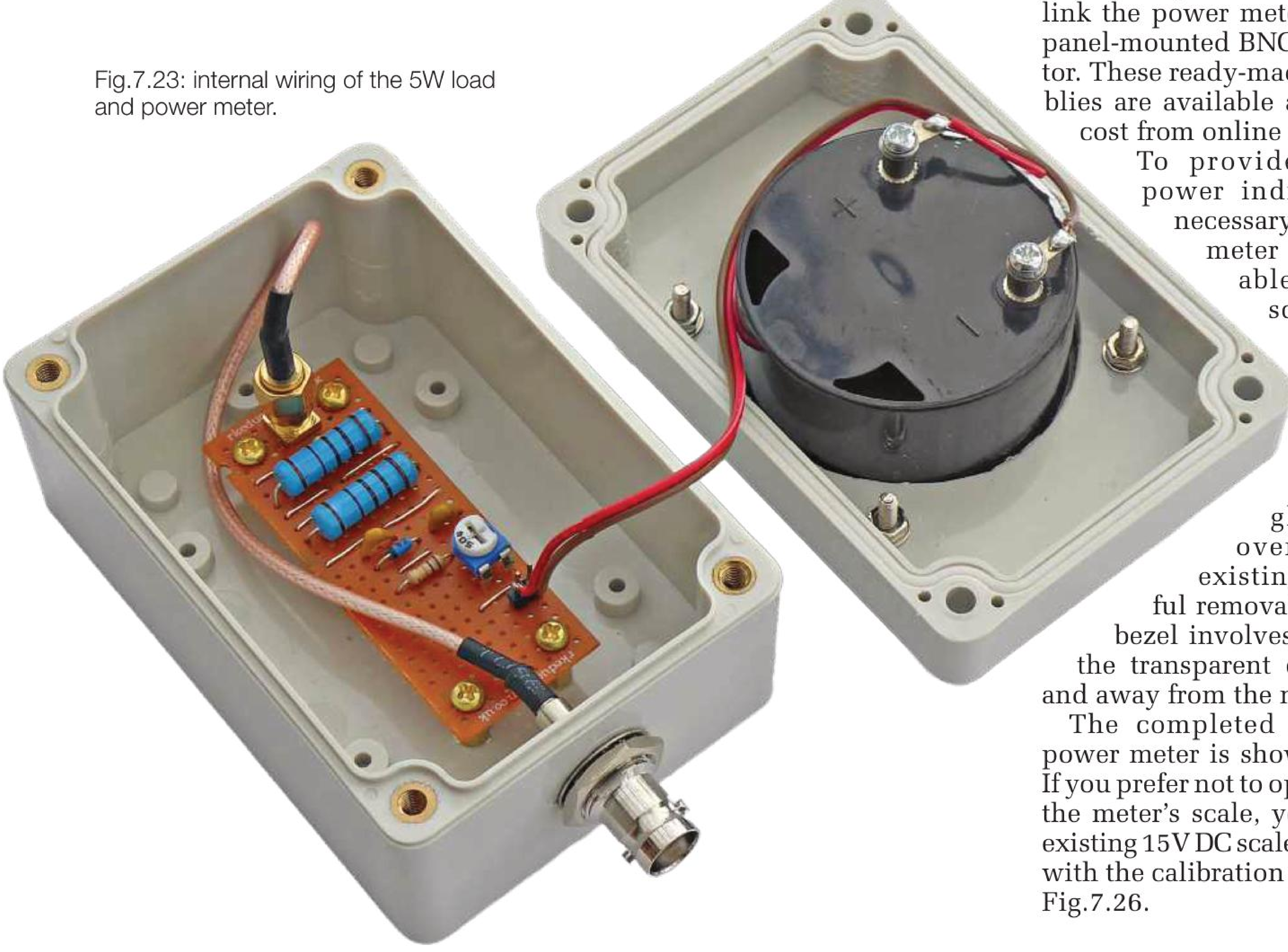
When the circuit board is complete (see Fig.7.22), it should be mounted in a suitable enclosure and interconnected as shown in Fig.7.23. When positioning the circuit board, adequate clearance must be left above it to accommodate the meter movement.

To avoid the somewhat tedious task of manually fitting the SMA and BNC connectors, we recommend the use of ready-made coaxial cables to link the power meter board to the panel-mounted BNC input connector. These ready-made cable assemblies are available at a reasonable cost from online suppliers.

To provide convenient power indications, it is necessary to replace the meter scale. A suitable replacement scale is shown in Fig.7.24. This can be copied and printed at a size that will allow it to be glued directly over the meter's existing scale. Careful removal of the meter's bezel involves gently prising the transparent cover upwards and away from the meter's body.

The completed 5W load and power meter is shown in Fig.7.25. If you prefer not to open and replace the meter's scale, you can use the existing 15V DC scale in conjunction with the calibration chart shown in Fig.7.26.

Fig.7.23: internal wiring of the 5W load and power meter.



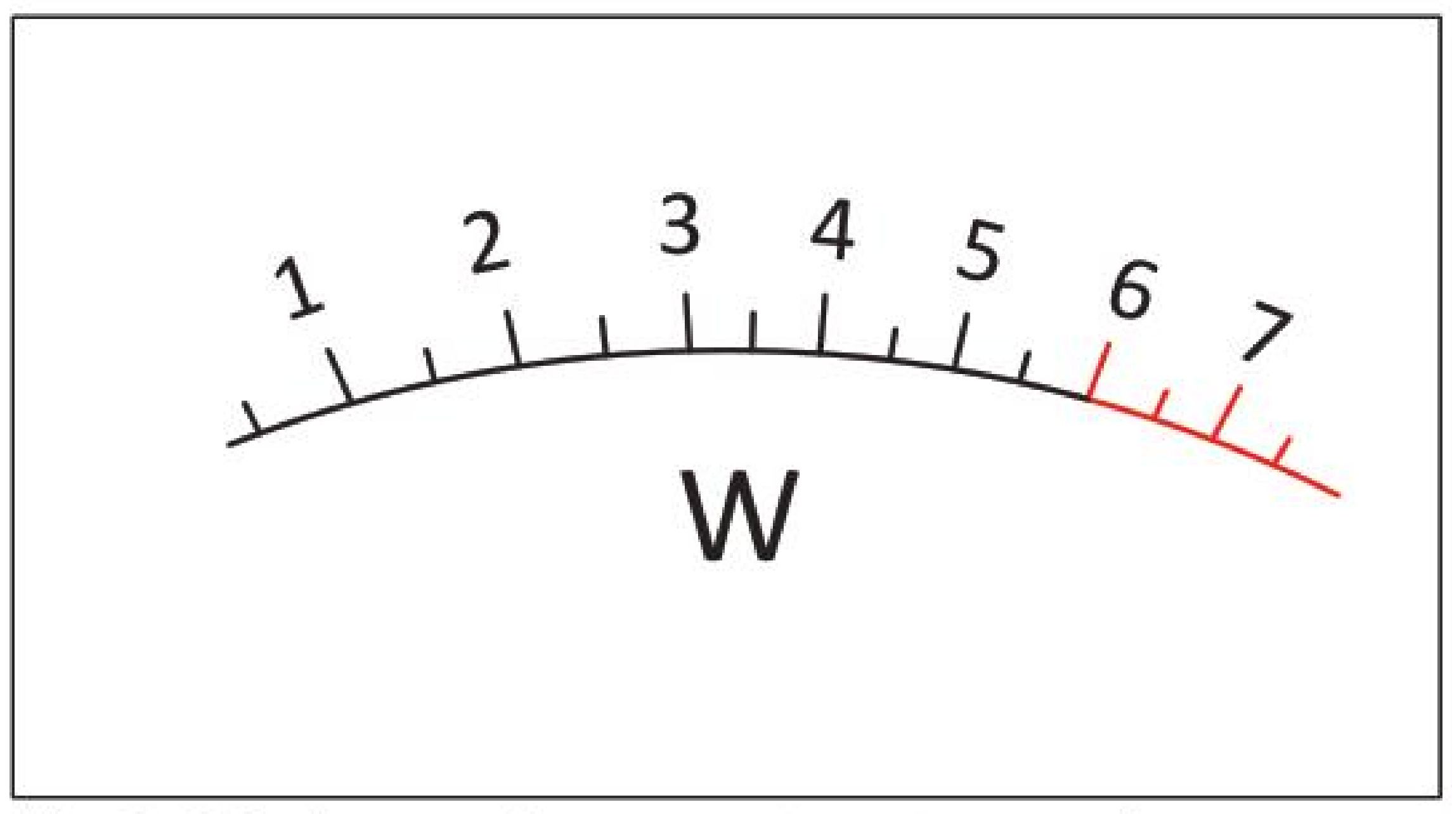


Fig.7.24: the replacement meter scale.

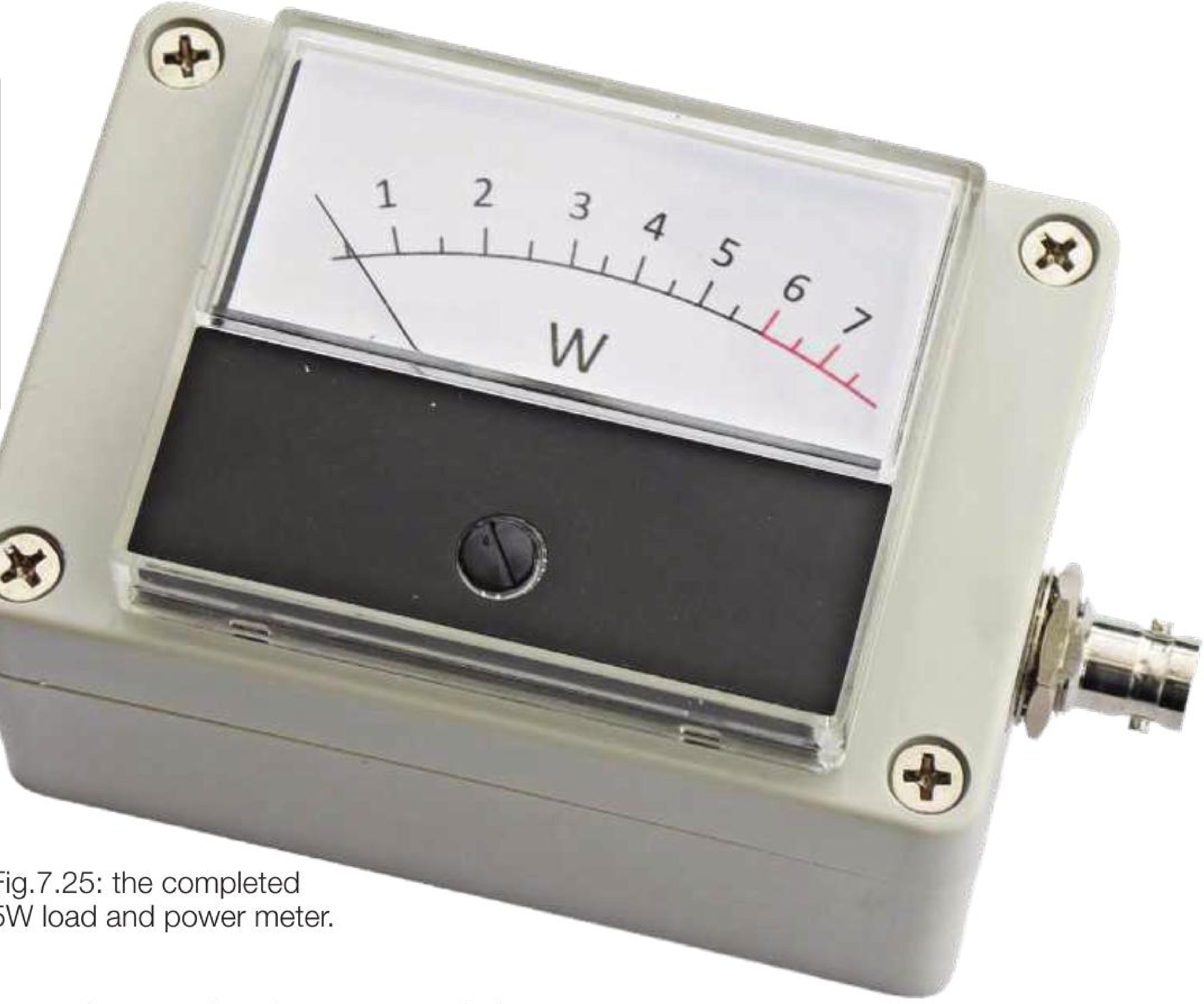


Fig.7.25: the completed 5W load and power meter.

### Calibration and testing

Calibration of the power meter is straightforward and simply requires an RF source capable of delivering 5W at a frequency between 3MHz and 10MHz. If you have an RF power meter at hand, you can use it as a reference. Simply adjust RV1 for identical readings.

Alternatively, the power meter can be calibrated using an RF voltmeter—including the RF probe featured in last month's *Hands-On* project. Connect the voltmeter across R1 or R2 and adjust the RF source for an indicated voltage of 15.8V RMS with a power of 5W applied (this power will be shared equally between the two parallel-connected load resistors).

Now carefully adjust RV1 so that the power meter scale reads exactly 5W if you've replaced the scale, or 10V if you've decided not to replace it. This completes the calibration process; the instrument is ready for use.

The power meter exhibits a VSWR of 1.05 at 7MHz. This represents a reasonably good impedance to a 50Ω source. However, due to the stray reactance associated with the simple construction method used, the load will become increasingly imperfect as the frequency of the applied power increases.

We recommend that the power meter should be limited to a maximum of 30MHz, at which point the VSWR will be in the region of 1.2. S-parameter data imported into AppCAD (see earlier) yields the S11 VSWR plot shown in Fig.7.27. This clearly shows the rapid rise in VSWR above 10MHz.

### Derating

All practical loads require derating. Theoretically, the two 3W resistors (R1 and R2) in this project should be capable of dissipating 6W. Unfortunately, it's necessary to progressively derate an RF load resistor because, in practical use, resistors are unable to dissipate their maximum rated power indefinitely without overheating (without forced airflow).

When subjected to continuous high power, such as the 6W theoretical maximum for the resistors used in this project, the resistors will become excessively hot if operated at this level for extended periods.

With all types of load, excessive heat can lead to premature failure, changes in resistance value, or even damage to surrounding components. Therefore, derating ensures reliable operation by reducing the allowable power dissipation as operating time or ambient temperature increases, preventing thermal stress and maintaining the working life of the components.

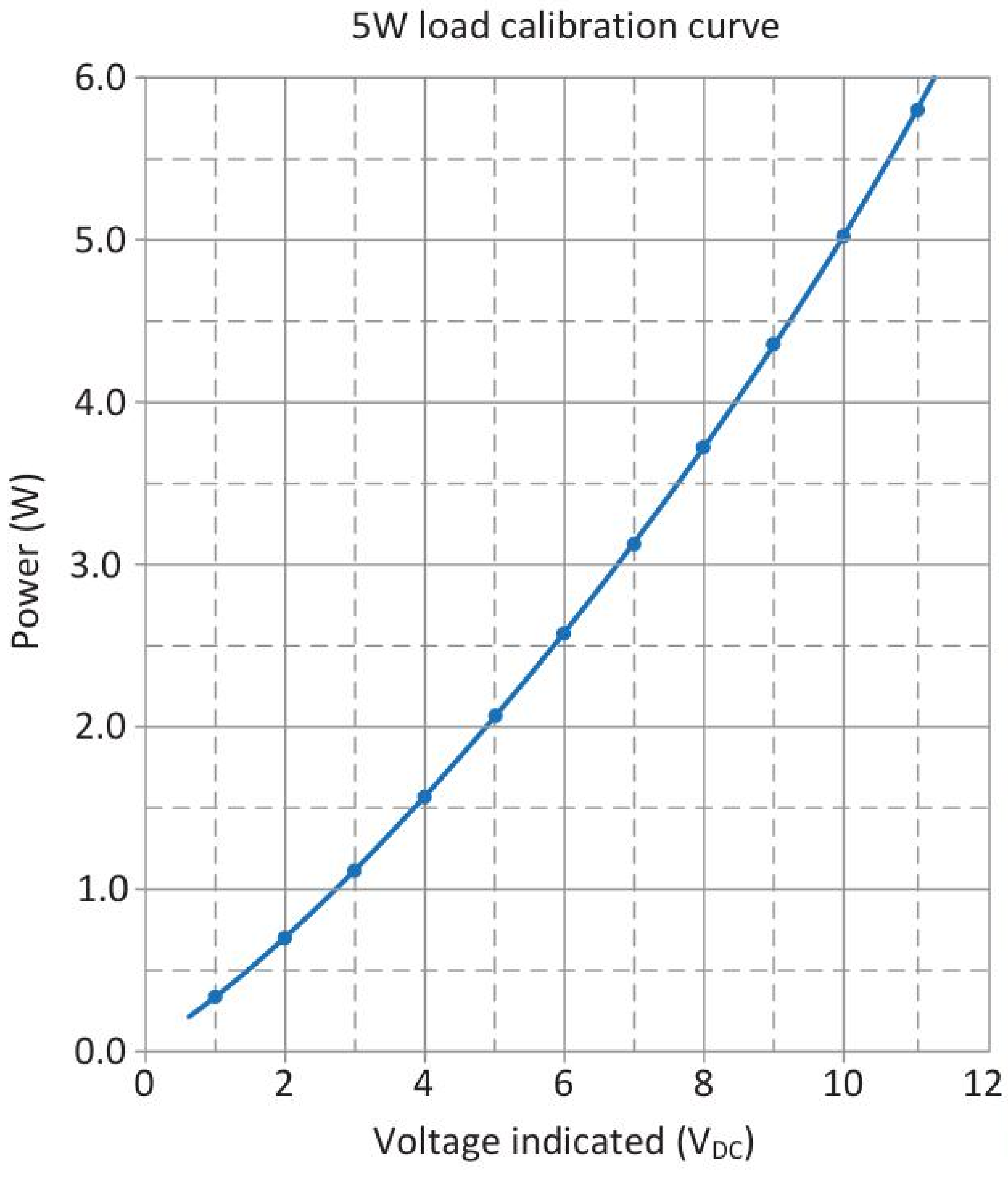
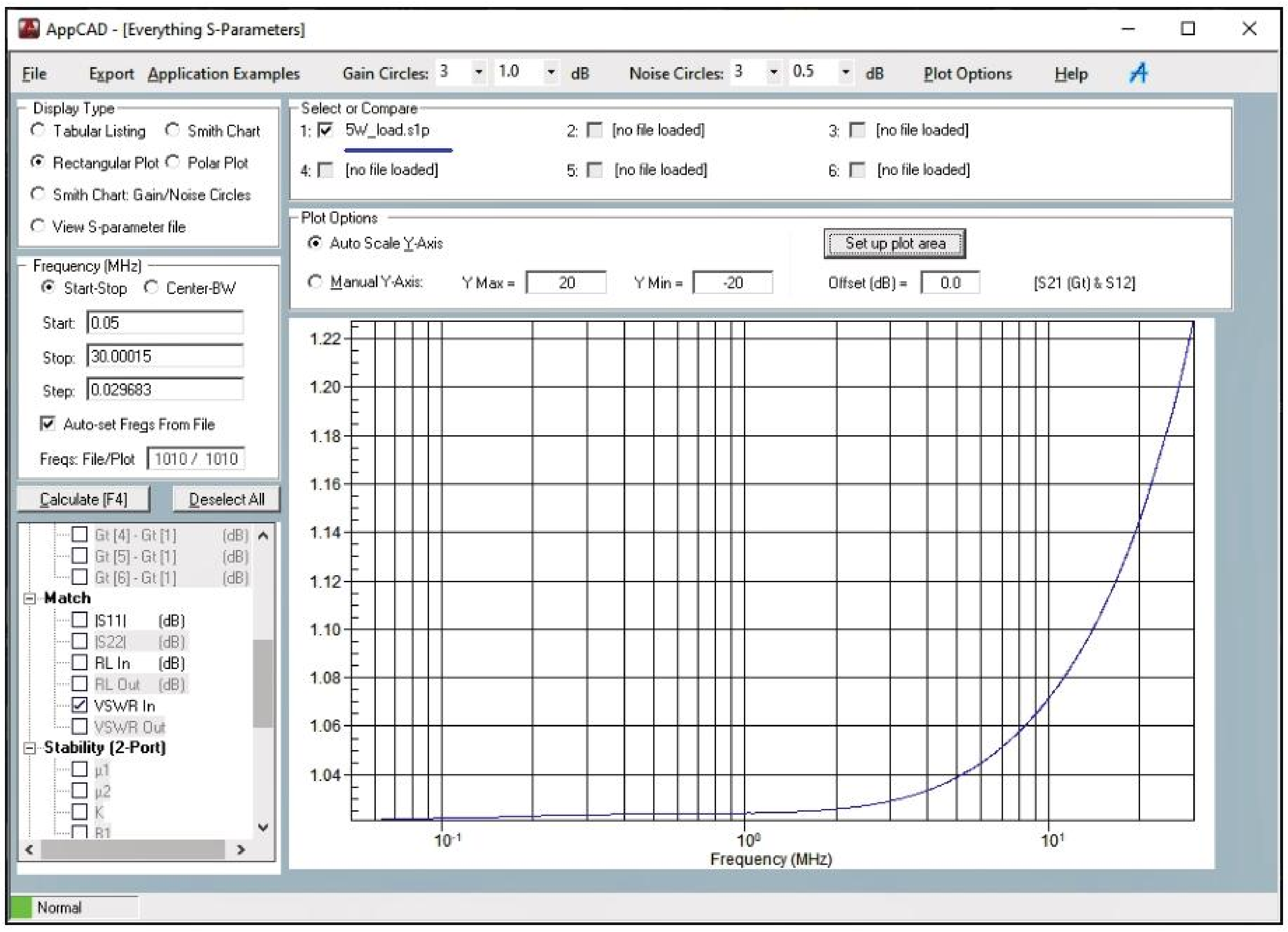


Fig.7.26: the 5W load & power meter calibration chart.



Gain Circles: 3 1.0 dB
Noise Circles: 3 0.5 dB

Display Type
○ Tabular Listing ○ Smith Chart
● Rectangular Plot ○ Polar Plot
○ Smith Chart: Gain/Noise Circles
○ View S-parameter file

Select or Compare
1: ☑ 5W_load.s1p
2: ☐ [no file loaded]
3: ☐ [no file loaded]
4: ☐ [no file loaded]
5: ☐ [no file loaded]
6: ☐ [no file loaded]

Plot Options
● Auto Scale Y-Axis
○ Manual Y-Axis: Y Max = 20 Y Min = -20 Offset (dB) = 0.0 [S21 (Gt) & S12]

Frequency (MHz)
● Start-Stop ○ Center-BW
Start: 0.05
Stop: 30.00015
Step: 0.029683
☑ Auto-set Freqs From File
Freqs: File/Plot 1010 / 1010

☐ |S11| (dB)
☐ |S22| (dB)
☐ RL In (dB)
☐ RL Out (dB)
☑ VSWR In
☐ VSWR Out
☐ µ1
☐ µ2
☐ K
☐ B1

Fig.7.27: an AppCAD VSWR plot from S11 data for the 5W load & power meter.

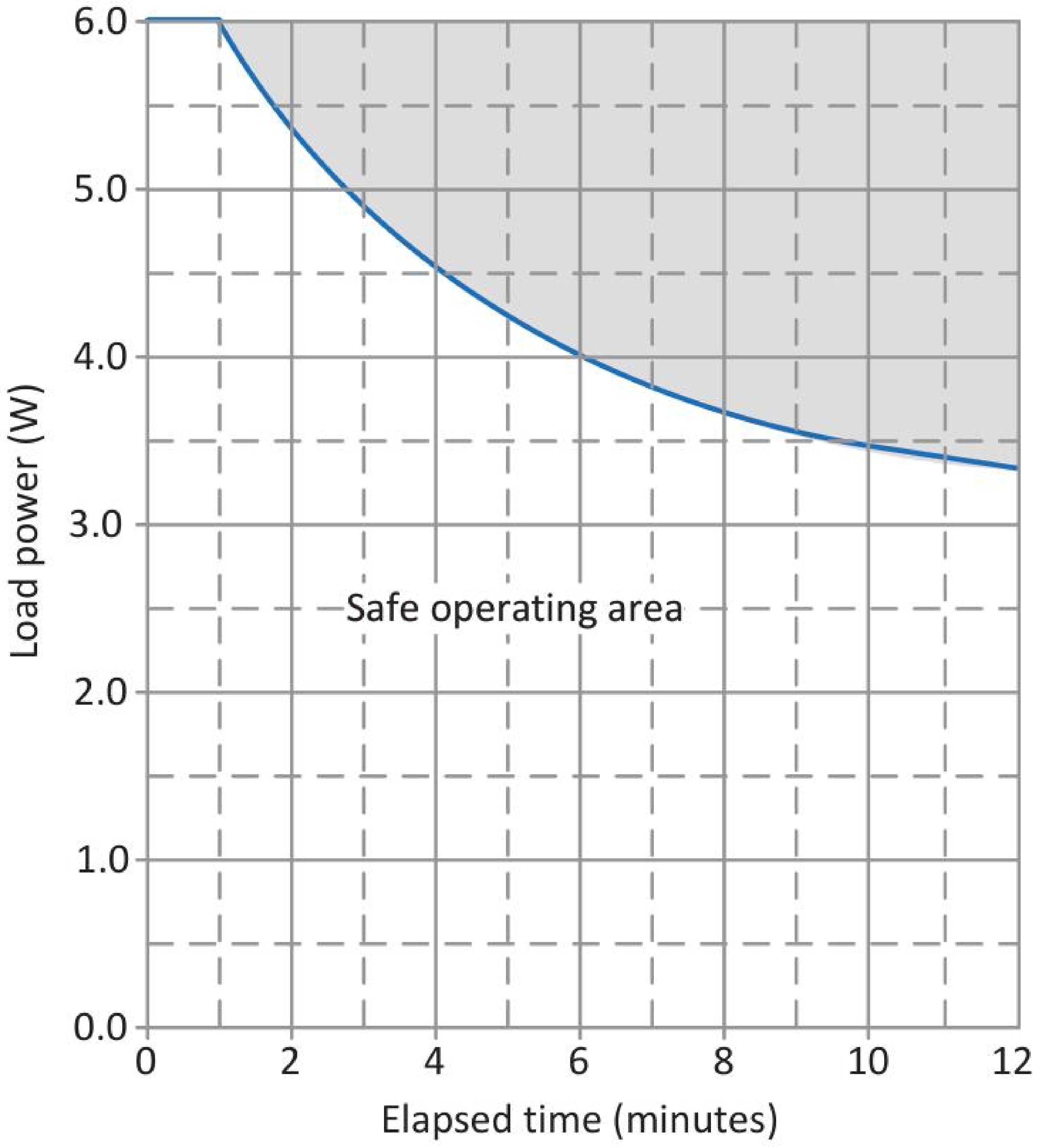


Fig.7.28: the recommended derating curve for the 5W load.

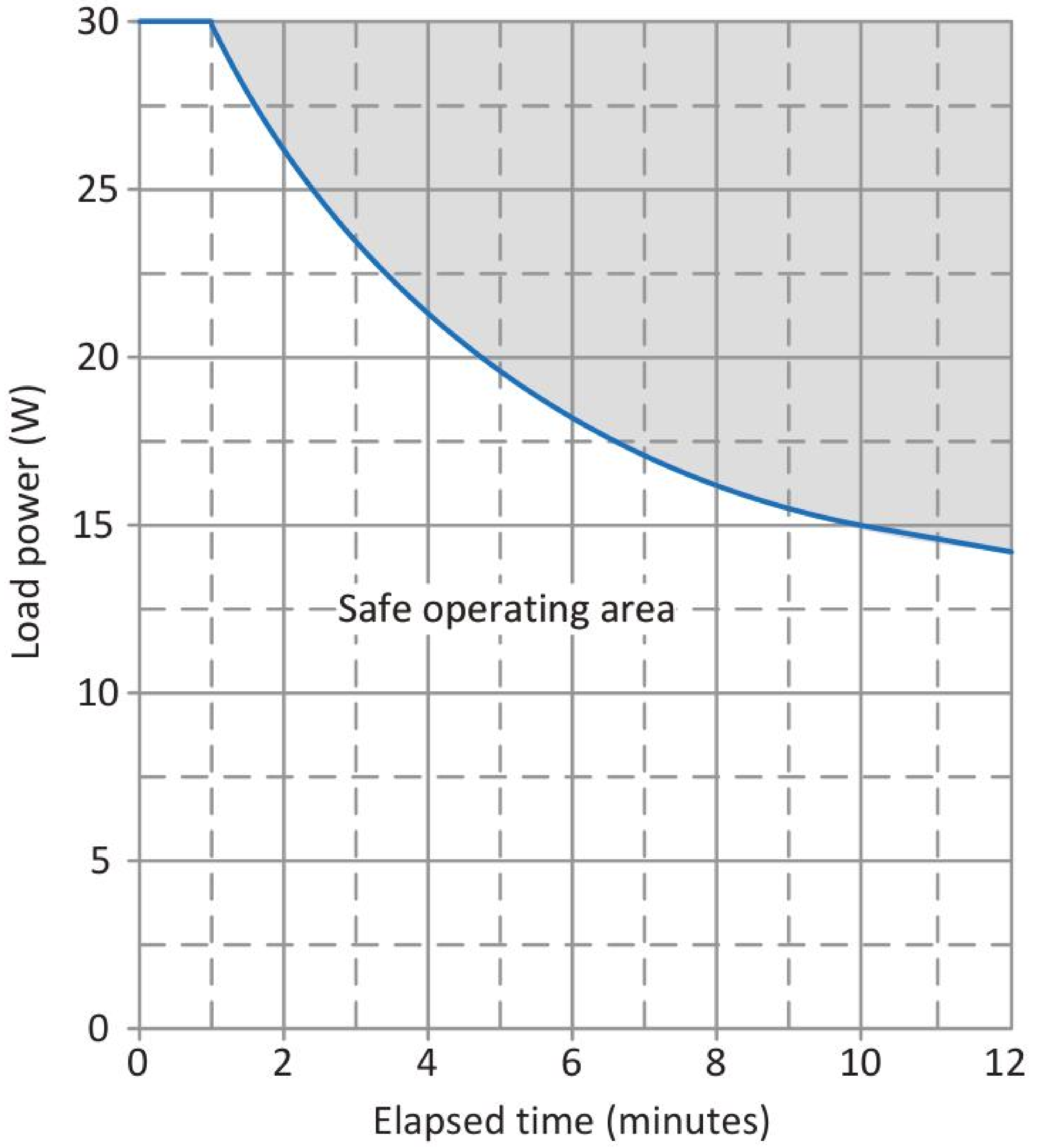


Fig.7.29: the recommended derating curve for the 25W load.

Our derating curve shown in Fig.7.28 indicates recommended safe operation limits: 6W (the maximum rated power) for up to one minute, 4W for four minutes and 3.5W for ten minutes. Just remember to keep within the 'safe operating area' marked on the chart!

### *Hands-On*: A 25W load

If you require a load with a better rating (for example, to test VHF transceivers), we have a second *Hands-On* project to put this right. This improved load is conservatively rated at 25W and is suitable for operation up to 450MHz.

The circuit of the 25W load is shown in Fig.7.31. The load uses a single 50Ω 100W RFP1109 'microwave resistor' mounted in a substantial diecast enclosure. This not only provides effective screening for the load but also provides a very effective means of dissipating the generated heat.

If necessary, to cater for higher powers (up to 50W), the enclosure can be augmented with one or more external heat dissipators. Suitably rated 50Ω 100W 'microwave resistors' are available from several online suppliers.

### Assembling the 25W load

The load resistor should be mounted as close to the SO-239 connector (SK1) as possible, as shown in Fig.7.30. Heat dissipation will be further improved by the application of a small amount of thermal paste between the resistor's metal mounting lugs and the inner surface of the enclosure.

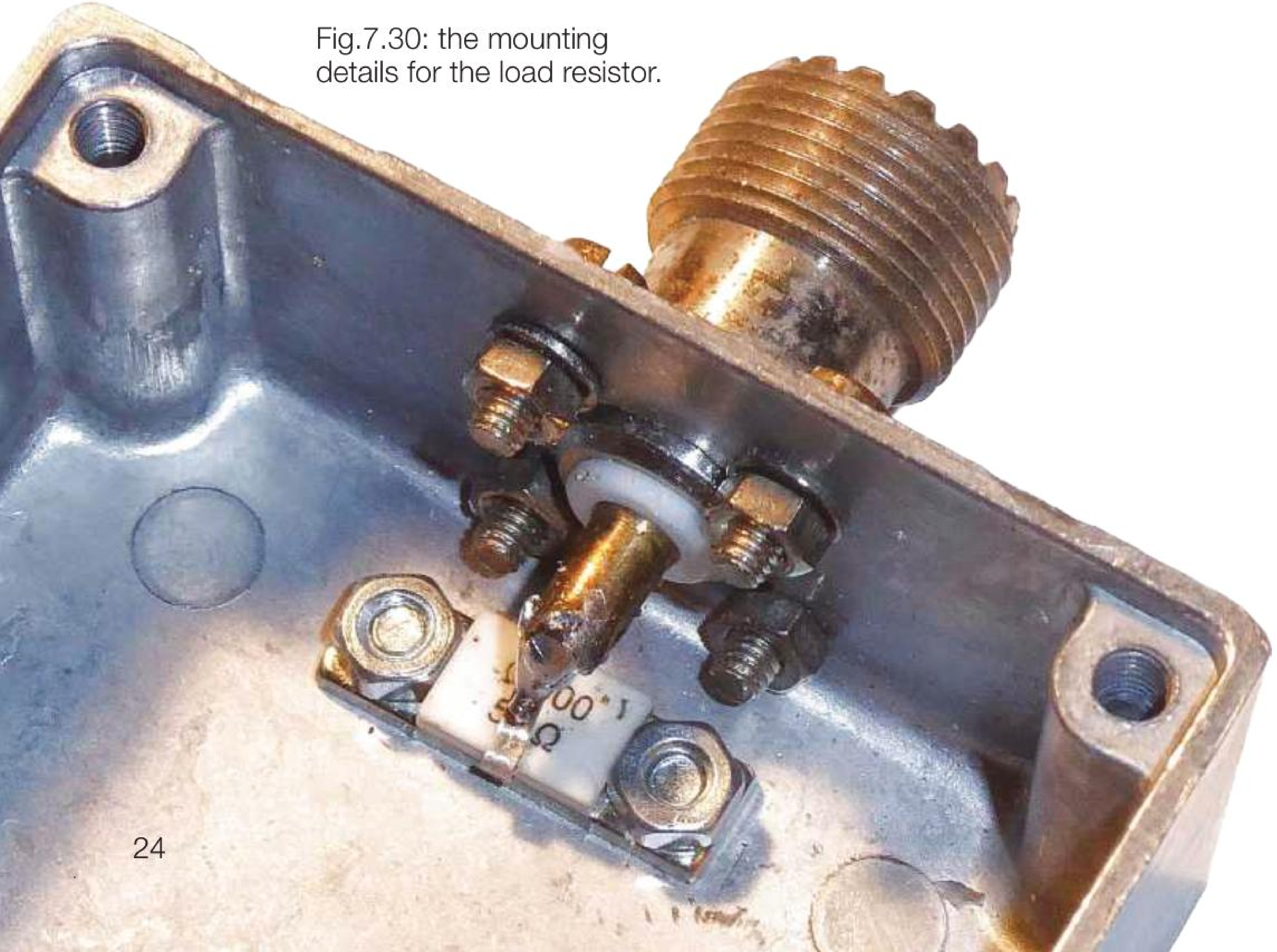


Fig.7.30: the mounting details for the load resistor.

Solder the resistor directly to the centre pin of SK1, but avoid applying excessive force when bending the resistor's connecting strip.

### Testing and derating

If an inline VSWR/RF power meter is available, it should be used to check the 25W load using an arrangement like that shown in Fig.7.32. The VSWR should be less than 1.05 at all frequencies up to 450MHz, and the load should be capable of sustaining a power of 30W for short periods and 12.5W indefinitely.

If required, a further heat dissipator can be bolted to the exterior of the enclosure. This should ideally be fitted near the mounting bolts fitted to R1 and as close to the load resistor as possible.

Our recommended derating curve for the 25W load is shown in Fig.7.29. This chart assumes that no external heat dissipator has been fitted. We recommend a maximum of one minute when 30W is applied, 20W for five minutes and 15W for ten minutes.

The prototype was placed on test with 15W applied for several hours at an ambient temperature of 20°C. At the end, the surface temperature

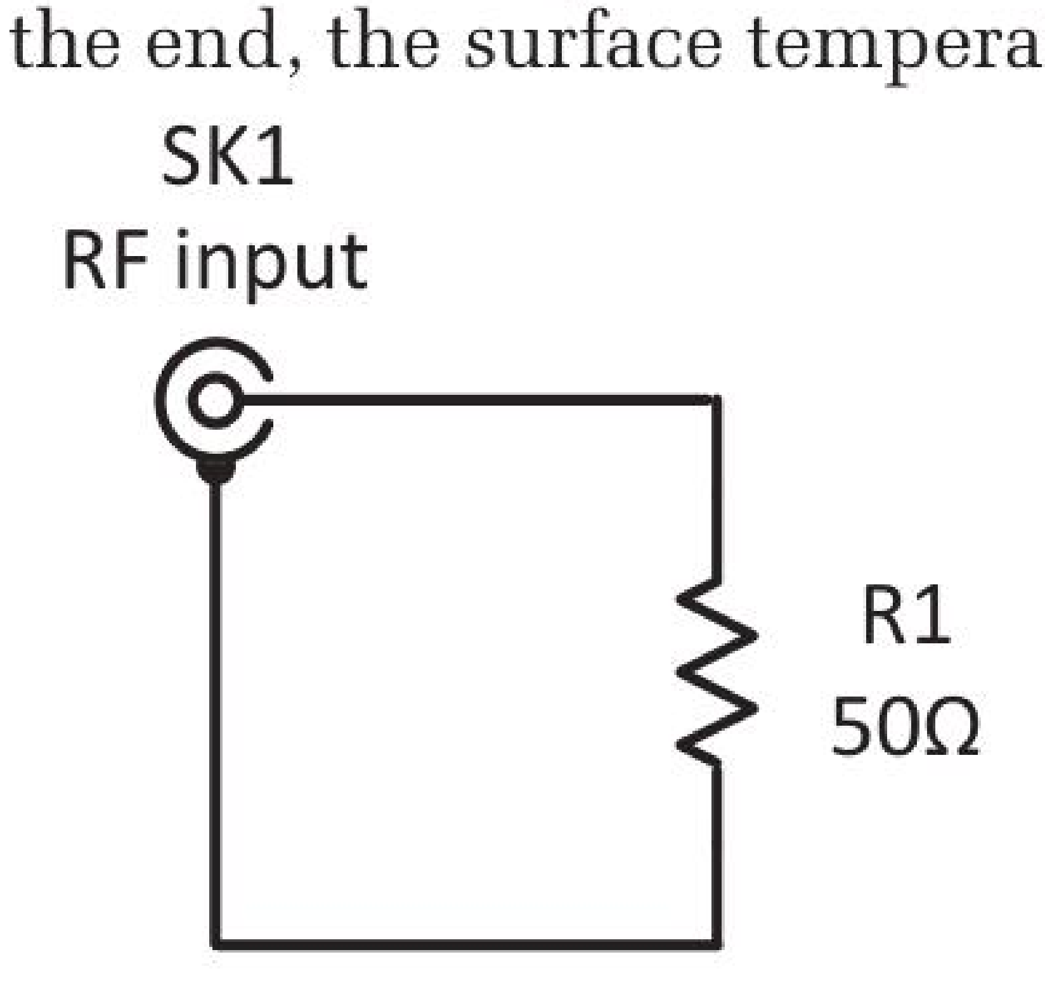


Fig.7.31: the circuit of the 25W load.

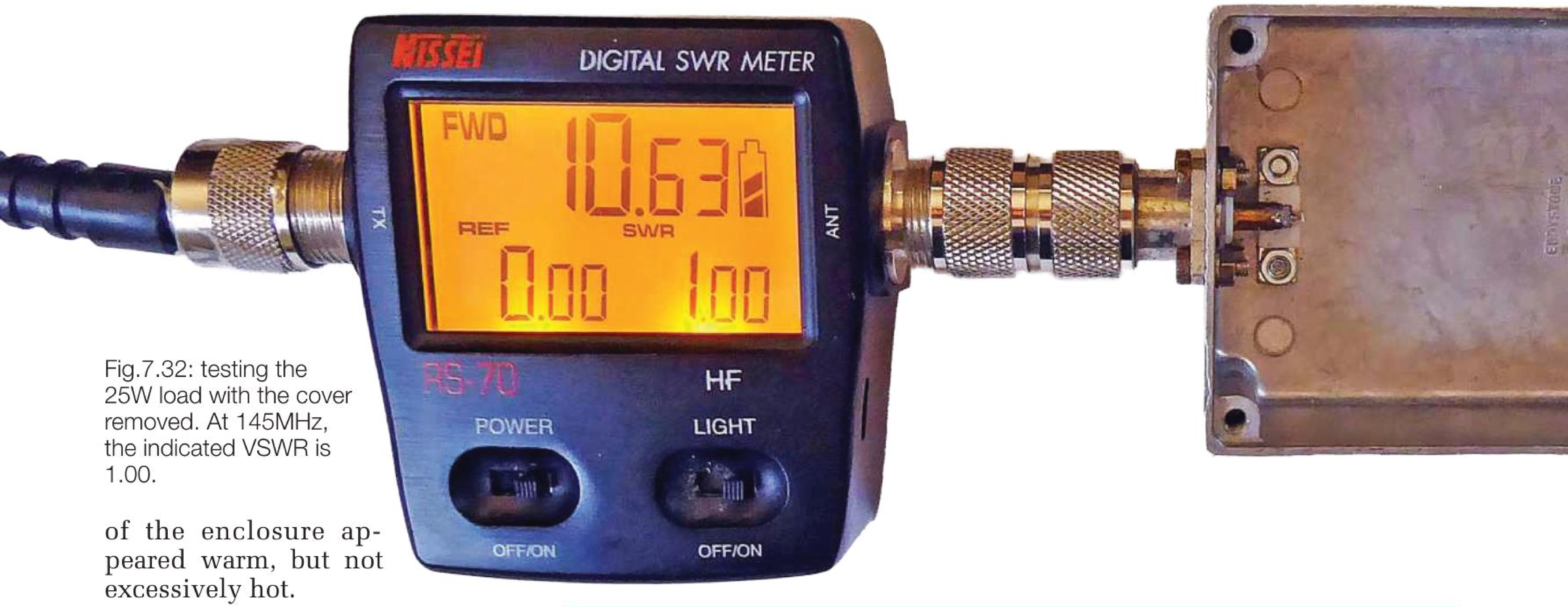


Fig.7.32: testing the 25W load with the cover removed. At 145MHz, the indicated VSWR is 1.00.

of the enclosure appeared warm, but not excessively hot.

**Coming up!**

Part 8 will introduce frequency changing methods, exploring multipliers and mixers with practical examples for testing in the popular Tina simulation package. Our *Hands-On* constructional project will be a 50MHz digital frequency meter. See you then! **PE**

## Parts List – 25W load

1 diecast enclosure, 110 × 60 × 30mm
1 SO-239 female coaxial connector (SK1)
1 RFP1109 (or similar) 50Ω 100W resistor
2 M3 countersunk head machine screws
2 M3 hex nuts and shakeproof washers
1 small tube or packet of thermal paste

## Parts List – 5W load and power meter

1 small ABS enclosure, 100 × 68 × 50mm or larger
1 25 × 64mm piece of stripboard (9 × 24 holes)
1 BAT42 Schottky diode (D1)
1 2-pin male 0.1in/2.54mm header (P1)
150kΩ miniature preset potentiometer/'trimpot' (RV1)
1 PCB-mounting female SMA connector (SK1)
1 15cm panel-mount BNC female to SMA male coaxial cable adaptor
1 15V DC analog voltmeter with zero adjuster **[eg, Velleman 7015]**
4 brass or nylon M3 × 10mm hex tapped spacers
4 M3 panhead machine screws
4 M3 countersunk head machine screws

**Capacitors**
1 10nF (C2)
1 100pF (C1)

**Resistors (all ¼W axial, ±5% or better)**
1 10kΩ ¼W axial, 5% or better (R3)
2 100Ω 3W axial resistors (R1, R2)

***The Fox Report**, continued from page 13...*

... radio web pages and the Sounds App) use AAC-LC coding at a range of qualities up to 320kbps. However, Freeview and Freesat max out at 192kbps, and DAB reduces rates to the best cases of 160kbps or 192kbps for Radio 3.

So what data rates does Freely use for BBC Radio? Netgem replied to my question with the question: "Please can you clarify what you mean by data rates?" After clarifying, I heard nothing back on this from Netgem and Freely simply ignored my questions on data rates.

Fortunately, Manhattan was on the ball and clarified, "All four national BBC radio stations use the same 320kbps high-quality stream on Freely....by default."

I also asked about the delay viewers and listeners will experience when switching between streamed channels, now and in the future when more and perhaps higher quality channels are added.

Says Manhattan, "It does vary a little depending on circumstances, but we aim for around two seconds maximum."

Says Netgem, "The target delay between Freely TV channels is three seconds. For radio channels it's five seconds: three seconds to load, and then two seconds to load the BBC Sounds App by pressing the Red Button."

Adding more TV and radio channels "will not impact the target delay times", says Netgem.

Freely acknowledges the need for "enhancements specifically aimed at boosting performance on lower-specification hardware" but will give no indication of what these "enhancements" are or what the target delays are, now or in the future.

Freely has pledged to "futureproof live TV for the streaming age", so I asked if Freely can assure people who buy first-generation receivers that they will not suffer increased streaming channel switching delays as more channels are added:

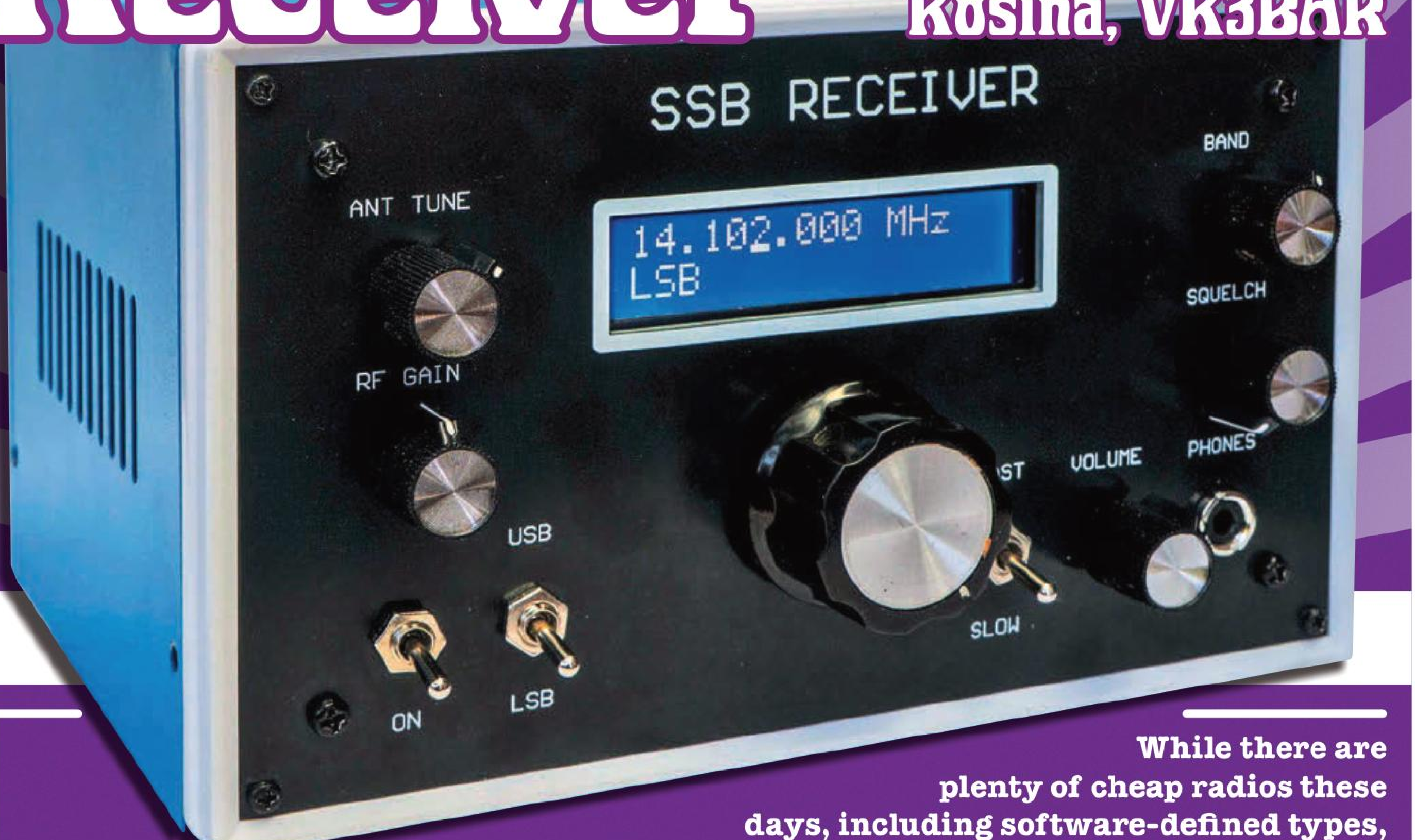
*We continue to evolve the channel tune speed across all of our Freely device...As part of these efforts, we recently completed a series of enhancements specifically aimed at boosting performance on lower-specification hardware, bringing all Freely viewing experience in-line.*

*We're also finalising work to improve overall UI [user interface] responsiveness across Freely, with specifically identified opportunities to optimise channel tuning further. Freely is continually undergoing improvements and enhancements, with regular updates throughout the year.*

*The number of channels on the Freely streaming platform is evolving and growing at pace. Freely viewers can also enjoy new and enhanced features added throughout the year. Over time, the platform is enhanced and the viewing experience will only be improved... The viewing experience will only be improved as more channels are added.* **PE**

# SSB Shortwave Receiver

### Part 1 by Charles Kosina, VK3BAR



**While there are plenty of cheap radios these days, including software-defined types, I decided to build this analog shortwave radio for the satisfaction of making it myself. I learned a lot about shortwave, SSB and how radios work in the process, which you will not get just buying an 'appliance'!**

**Radio receiver architectures have changed dramatically** in the last few years. Digital techniques have largely displaced the analog techniques from the past. Radio receivers are now available at ridiculously low prices from various internet sources.

The simplest ones are the Software Defined Radios (SDR) that are a small module that plugs into a USB port. The typical coverage is from 100kHz to 2GHz; they rely on the processing power of the attached computer to recover the desired signal.

They are not 'communications receivers', as their noise figure and immunity from intermodulation are quite poor. With no input tuneable filter, a strong signal can easily overload the front-end circuitry. But for many, the performance is quite adequate. The screen "waterfall display" showing signals is very useful.

Still, it isn't too hard to build an analog shortwave receiver with decent performance, as I shall explain shortly.

## Performance

The performance of this unit is quite reasonable. I set my signal generator to 1µV (-107dBm) output and the signal-to-noise ratio was about 13dB for most of the range (slightly less at 30MHz).

Inserting a 20dB attenuator gave me a signal of about 0.1µV (-127dBm) and it was still audible! At that level, communication via Morse Code (CW) would be possible, but it is too weak for SSB voice reception. Yes, there are the unavoidable birdies, but they do not interfere greatly.

What about on-air tests? The only HF antenna I have at present is an end-fed half-wave dipole on the 40m band (7MHz). This is fed with 50Ω coaxial cable and uses a 49:1 'unun' (unbalanced to unbalanced) transformer. The measured SWR (standing wave ratio) is between 1.2 and 1.3.

Unfortunately, the ambient noise here is rather high with all the electrical equipment in the surrounding houses, producing heaps of RF hash, so it needs a fairly strong signal to get through. Comparing this receiver with my Bando Technic 5D transceiver, the sensitivity is much the same.

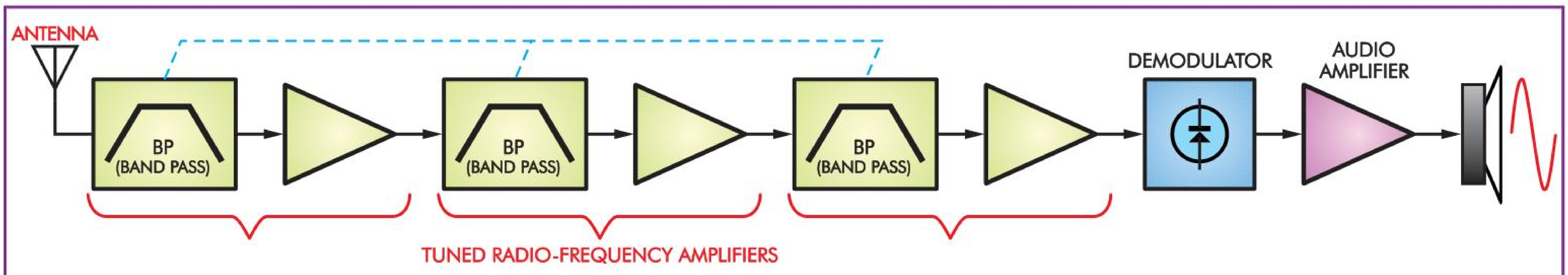

ANTENNA
BP (BAND PASS)
BP (BAND PASS)
BP (BAND PASS)
DEMODULATOR
AUDIO AMPLIFIER
TUNED RADIO-FREQUENCY AMPLIFIERS

**Fig.1: a TRF receiver comprises several identical RF amplifiers tuned to the same frequency. Most very early radios used this configuration.**

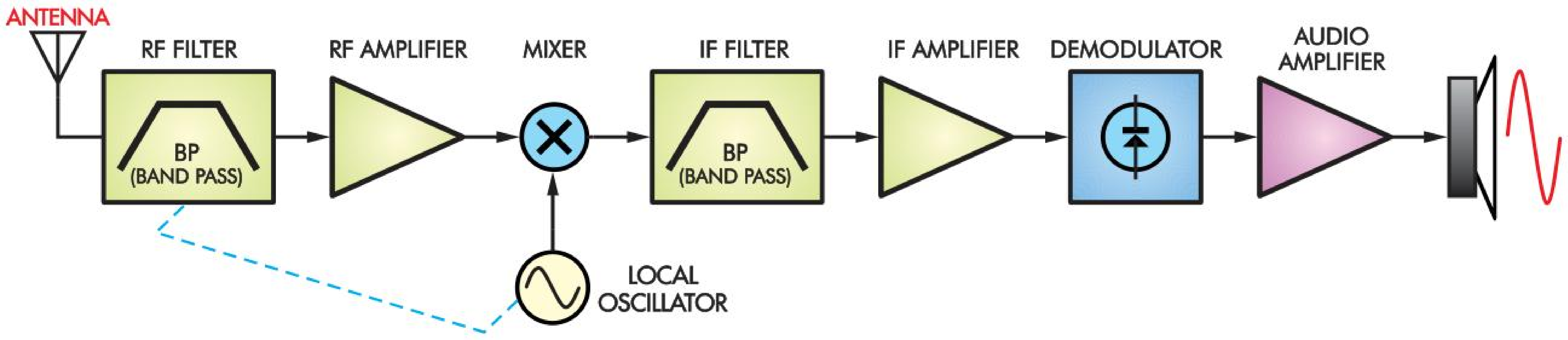

ANTENNA
RF FILTER
RF AMPLIFIER
MIXER
IF FILTER
IF AMPLIFIER
DEMODULATOR
AUDIO AMPLIFIER
BP (BAND PASS)
BP (BAND PASS)
LOCAL OSCILLATOR

**Fig.2: the superhet was an early game changer. By mixing the amplified, tuned incoming signal with an oscillator frequency that tracks above it, the signal is shifted down to a lower, fixed (intermediate) frequency. Signals at that frequency are easier to filter out and demodulate.**

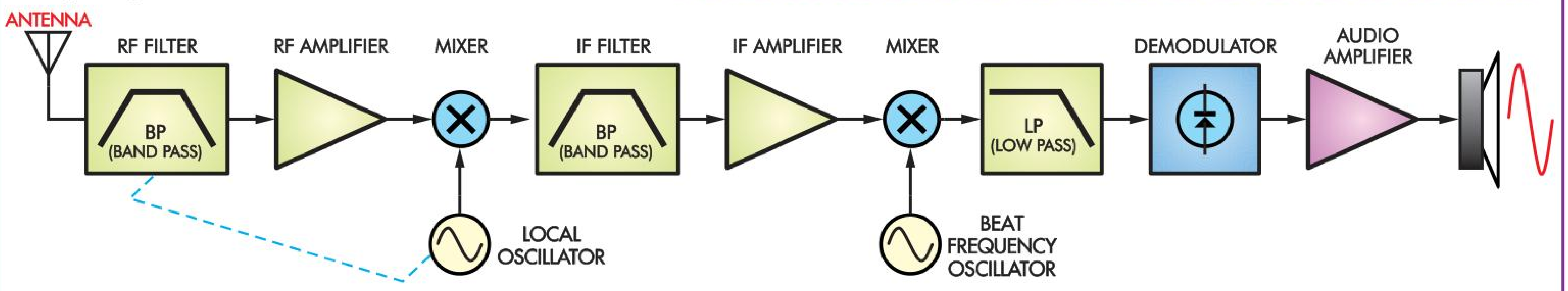

ANTENNA
RF FILTER
RF AMPLIFIER
MIXER
IF FILTER
IF AMPLIFIER
MIXER
DEMODULATOR
AUDIO AMPLIFIER
BP (BAND PASS)
BP (BAND PASS)
LP (LOW PASS)
LOCAL OSCILLATOR
BEAT FREQUENCY OSCILLATOR

**Fig.3: an SSB receiver is a bit more complex as it needs to operate without the carrier wave or half the signal spectrum. The modulated signal is recovered by mixing it with the output of the BFO in a second mixer stage.**

A list of the features and specifications for this receiver includes:

- Covers the shortwave band from 3MHz to 30MHz in two spans
- Sensitivity: 1μV (-107dBm) for a 13dB SNR (reception possible @ 100nV/-127dBm).
- 2.7kHz speech bandwidth
- Runs from 12V DC @ 500mA
- Digital tuning with frequency display
- Analog controls for tuning, volume, RF gain and squelch
- USB/LSB decoding
- Fast or slow AGC
- RSSI display
- Built-in speaker and headphone jack

## Radio receiver types

I will start with a brief summary of radio techniques over the last 100 years or so.

Back in the 1920s, a typical radio was most likely a TRF (tuned radio frequency) set. They consisted of several cascaded tuned amplifiers, each set to the same frequency, as shown in Fig.1. A detector extracted the audio signal at the end of the chain, which was then amplified to drive headphones or a loudspeaker.

The valves used were initially triodes, and with high feedback capacitances, were subject to instability, leading to oscillation. A neutralising system was developed to feed back a phase-shifted version of the signal, minimising or preventing this instability. Later, tetrode and pentode valves were used that had much lower feedback capacitance and obviated the need for neutralisation.

Not only were these early radios difficult to set up and use, but they lacked the selectivity to reject unwanted strong signals on different frequencies. However, by using regeneration, where a portion of the amplified signal is fed back to the input grid of the triode and pentode, much higher gain and selectivity could be achieved.

## Enter the superhet

The problems of the TRF receiver were largely overcome by the superheterodyne architecture. Edwin Armstrong is often credited with the invention of this technique, but others filed patents only months apart. Legal battles followed, and French engineer Lucien Lévy was awarded a patent that included seven of the nine claims in Armstrong's application.

Fig.2 shows the superheterodyne architecture. The incoming signal passes through a tuned filter, followed by an optional RF amplifier. Then follows the mixer, where the incoming signal is multiplied by another frequency from a local oscillator. The multiplication results in two extra signals, being the sum and difference of the frequencies.

For example, if the incoming signal is 1000kHz, and the local oscillator is at 1455kHz, the output from the mixer will contain signals at 455kHz (1455kHz – 1000kHz) and 2455kHz (1455kHz + 1000kHz). It will also include the original 1000kHz signal.

The following band-pass filter selects just the 455kHz portion, which is amplified by the IF (intermediate frequency) stage(s) and passed to a detector and audio amplifier as before. However, this technique is only suitable for receiving AM (amplitude modulated) signals.

To receive SSB (single sideband) broadcasts, discussed further below, an additional mixer stage is necessary after the IF amplifier(s). This mixes the IF signal with that from a BFO (beat frequency oscillator), and its output goes through a low-pass filter (LPF), as shown in Fig.3.

This architecture remained the normal way that radios were built for many decades. Most of the selectivity (ie, rejecting unwanted station signals) came from the IF filter.

However, consider a broadcast at 1910kHz with the common 455kHz IF. With the set tuned to 1000kHz, when mixed with the LO at 1455kHz, this signal also will produce a 455kHz output from the mixer. This is termed the image frequency, and it is why there is an input band-pass filter, to attenuate this image.

A single tuned circuit was adequate for the broadcast band frequency range of 530kHz to 1,600kHz, but once shortwave broadcasting became commonplace, the single tuned circuit was inadequate at frequencies above 3MHz and resulted in ‘double spotting’ of the same input signal.

This was often tolerated, but to get around it, double- and triple-conversion superhet sets were used for better performance. This resulted in a higher-frequency initial IF signal, which was then mixed again to obtain a lower-frequency secondary IF signal.

There are some limitations to the superheterodyne architecture. Spurious signals are generated as a result of the mixing process or from non-ideal components (like harmonic distortion). Spurious signals may be produced by the oscillator, mixer, or other components in the receiver.

The local oscillator may generate harmonic signals that mix with the RF signal, producing unwanted spurious signals. In nonlinear systems, two or more signals can combine to produce additional unwanted frequencies, known as intermodulation distortion.

Some of these spurious signals are characterised by a rapid tuning rate. The whistle or chirp that is produced changes in frequency much faster than the tuning of the receiver. Hence, they were called “birdies”.

## Hartley phasing

There are alternatives to the superheterodyne receiver. A variation is to use the Hartley phasing method, as illustrated in Fig.4. The incoming signal (ωs) is fed into two mixers. The local oscillator is at the same frequency, and an RF phase shift network of 90° will mix with the incoming signal to produce two signals at baseband, but at 90° apart.

These signals are called I (in-phase) and Q (quadrature). The Q signal is applied to an audio phase shift network, which in mathematical terms is a Hilbert transform. This shifts the entire audio spectrum by 90°.

However, this arrangement is impractical. A better approach is using two separate phase-shift networks of +45° for the I signal and -45° for the Q signal, as shown in Fig.5. These are then summed and filtered to produce the demodulated signal. This is a simplified explanation of the phasing system; there are plenty of online references that give a detailed mathematical analysis.

The phasing method is elegant in its simplicity, but there are practical problems in its realisation. It is relatively easy to have an accurate 90° phase shift, but the audio phase shift network requires an extremely high precision in components to maintain the accurate phase shift over the whole range of frequencies.

This is why the analog method has been superseded by digital techniques. In current SDR receivers, the I and Q signals are sampled by analog-to-digital converters and the Hilbert transform is done by software. It does require a fast processor, as found in modern computers.

I decided to investigate if a phasing receiver was practical using an analog phasing network. There are designs available to implement the Hilbert transform in hardware, but it requires careful matching and selection of components, preferably to within 0.1%. One such design is shown in Fig.6, and I built a test module to test its practicality.

I bought about fifty of the 10nF

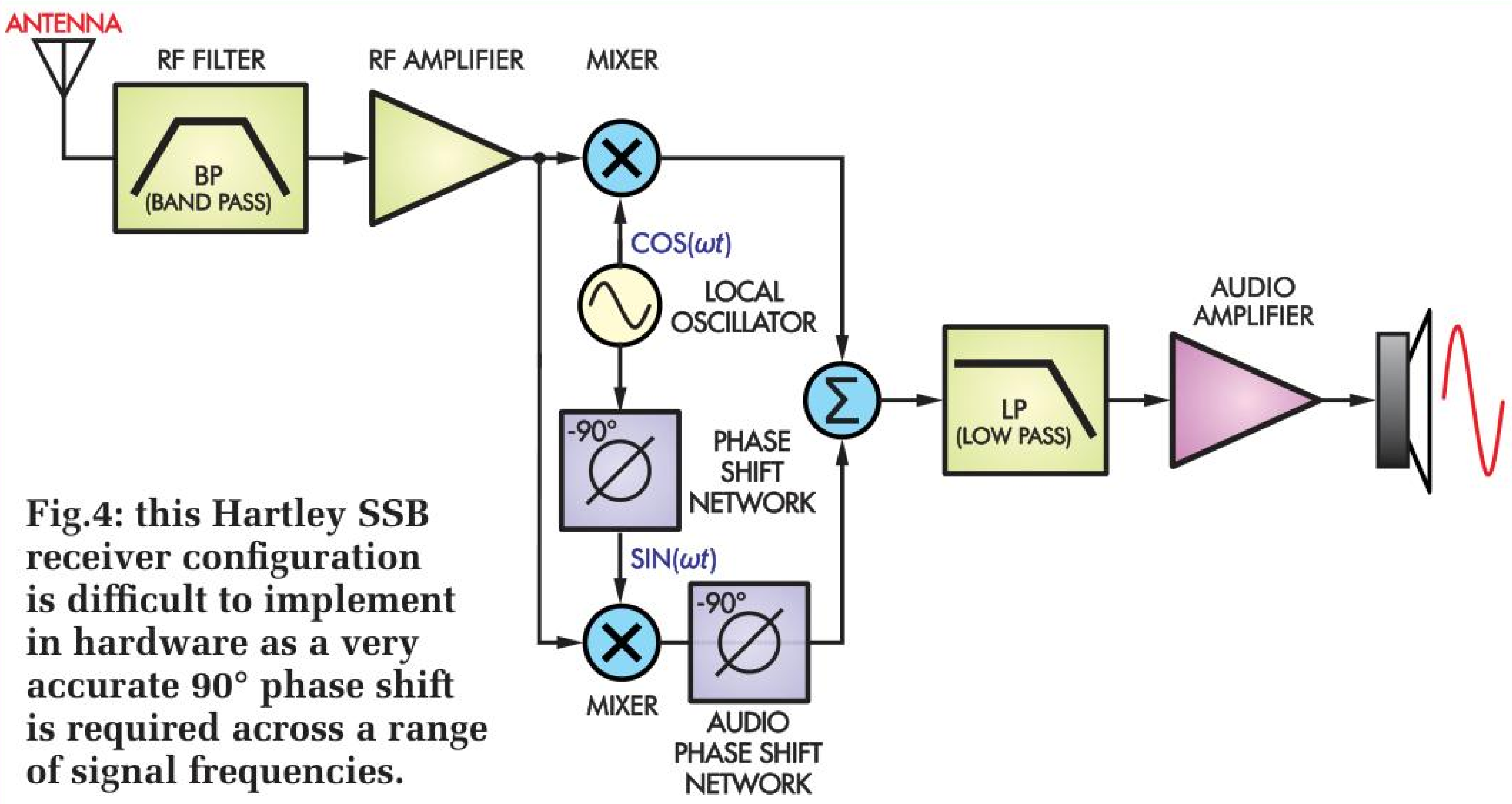


**Fig.4: this Hartley SSB receiver configuration is difficult to implement in hardware as a very accurate 90° phase shift is required across a range of signal frequencies.**

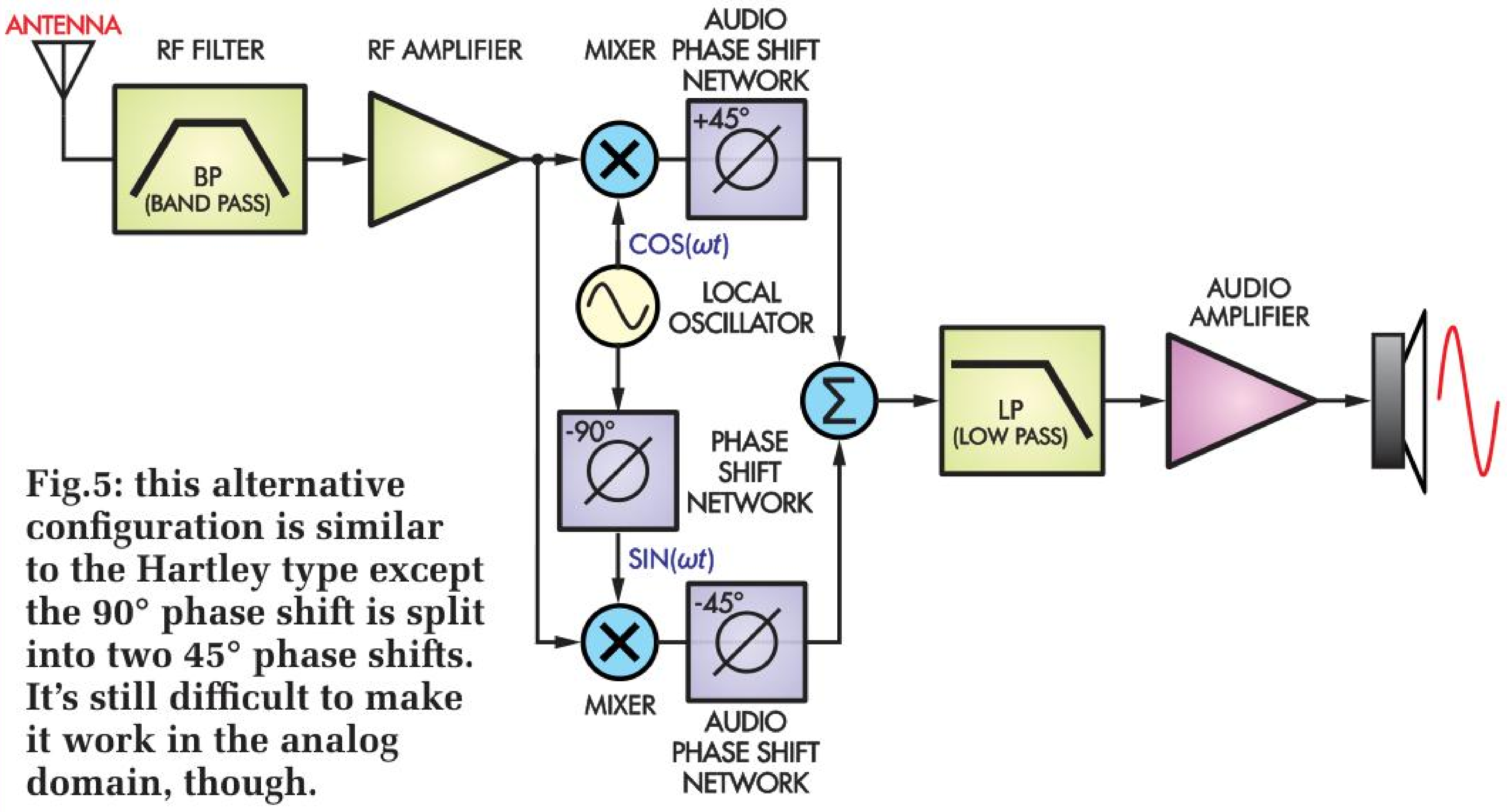


**Fig.5: this alternative configuration is similar to the Hartley type except the 90° phase shift is split into two 45° phase shifts. It’s still difficult to make it work in the analog domain, though.**

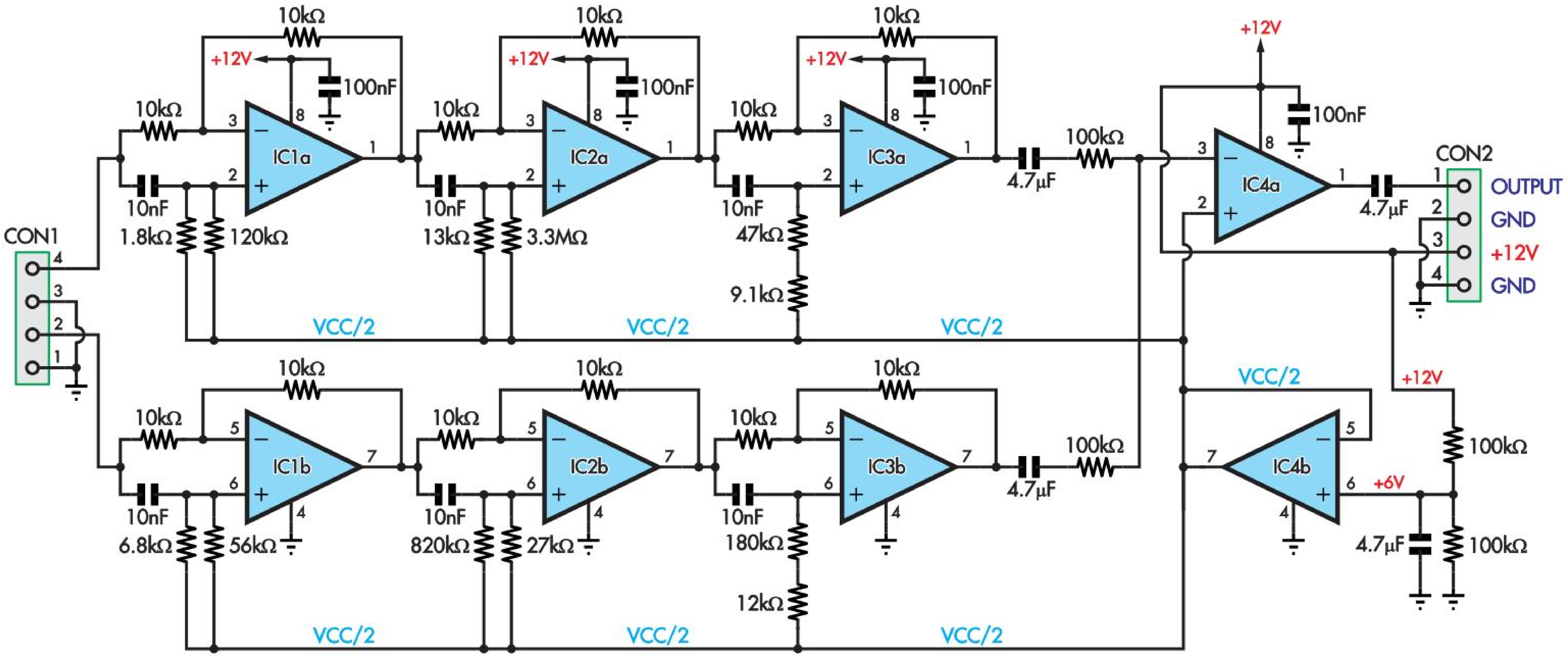

10kΩ
+12V
100nF
10kΩ
IC1a
IC2a
IC3a
IC4a
IC1b
IC2b
IC3b
IC4b
10nF
1.8kΩ
120kΩ
13kΩ
3.3MΩ
47kΩ
9.1kΩ
6.8kΩ
56kΩ
820kΩ
27kΩ
180kΩ
12kΩ
4.7μF
100kΩ
VCC/2
+6V
CON1
CON2
OUTPUT
GND
+12V
GND

Fig.6: an example of a phase shift network that provides a more-or-less fixed phase shift across a range of frequencies.

capacitors and, by measuring them to four-figure resolution, I selected a batch where all were within 0.1% of each other. The exact value is not quite as important as the matching.

I originally thought that 8-pin SIL (single in-line) resistor networks with four 10kΩ resistors each would be closely matched, but found that was not accurate enough. The alternative was to select from lots of 10kΩ SMD resistors for a matched set. The hardest part is getting the other six resistors to an exact value.

For example, a value of a 12,960Ω is needed, which is realised by two resistors in parallel, 13kΩ and 3.3MΩ. But this required measuring and selecting resistors that were close to the nominal value. Some values were difficult to get exactly.

Fig.7 shows the measured phase shift of my prototype which, while close to 90°, is not really close enough. The sideband rejection would be 40dB at best. Also, this is not the sort of design that can many readers would bother to build.

It is possible to eliminate the Hilbert transform; one solution is the Weaver architecture. Following the baseband low-pass filters, we have another pair of mixers, as shown in Fig.8.

The frequency injected into the second pair of mixers is at about half the bandwidth. Again, two signals 90° out of phase are needed, called the pilot tone. A further LPF extracts the wanted signal.

There are many articles and papers describing the Weaver method, many with quite complicated mathematics. It is described in detail at **https://pemag.au/link/ac51**

I built a receiver with this architecture, copying some of the design ideas available on the internet. After many hours of trying to get decent performance,

I eventually gave up. Getting the accuracy and balance between the I and Q channels just proved too hard.

Overall, the receiver was far too noisy; I suspect because of the multiple mixers, and I could not get rid of the pilot tone in the output. I would be interested to hear from readers who

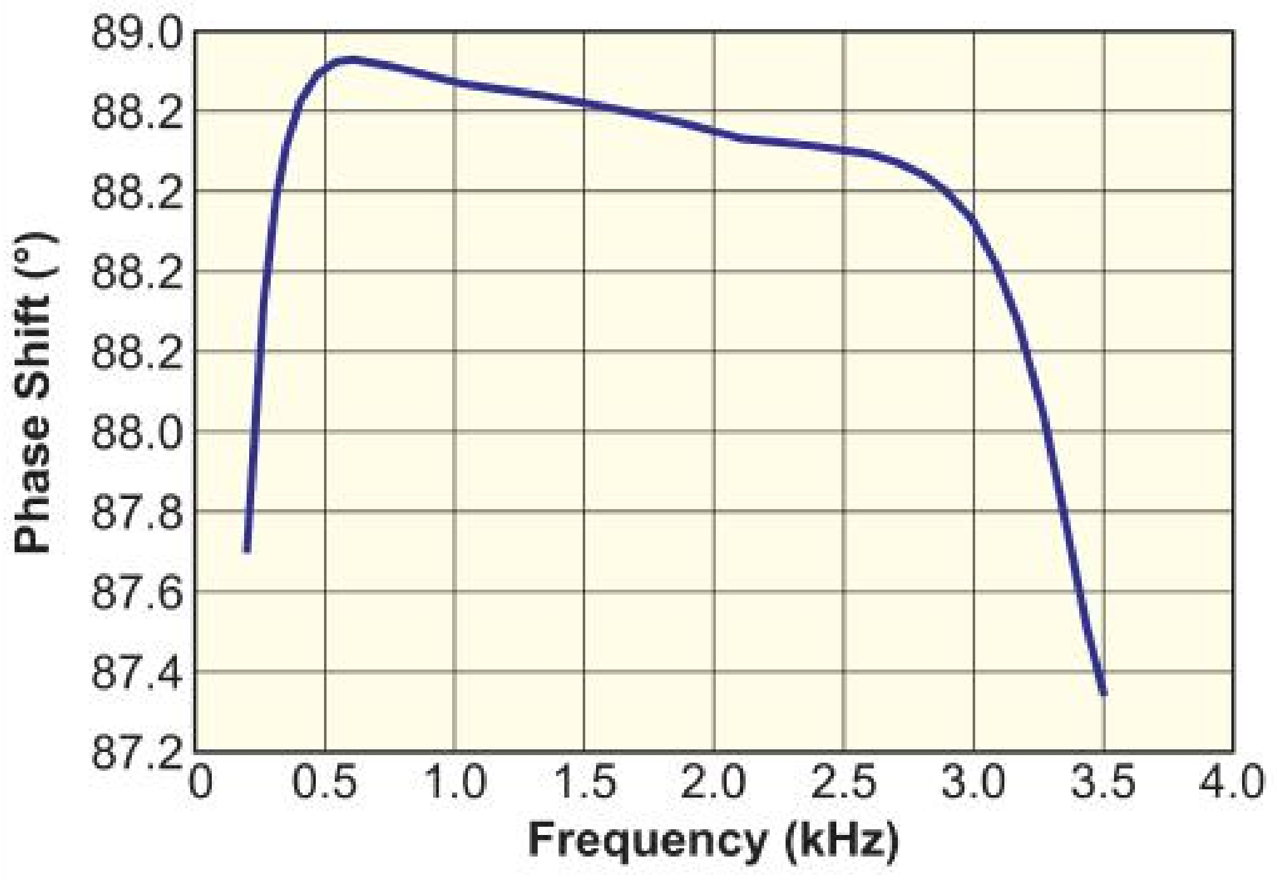


Fig.7: the performance of the phase shift network shown in Fig.6. Even with hand-selected matching components, it doesn't quite hit 90°, nor is it perfectly flat with frequency across the band of interest.

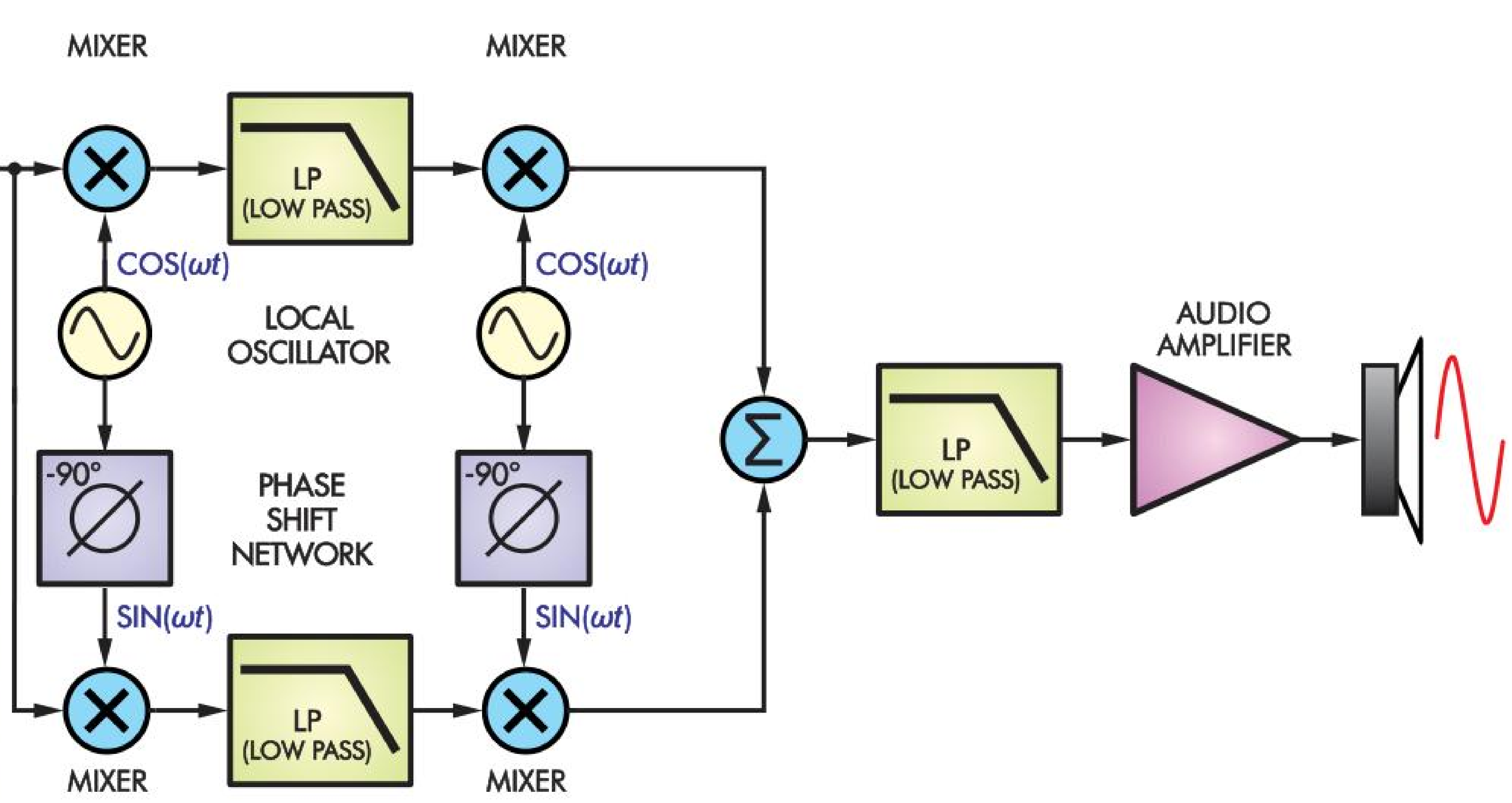

ANTENNA
RF FILTER
BP
(BAND PASS)
RF AMPLIFIER
MIXER
MIXER
LP
(LOW PASS)
COS(ωt)
COS(ωt)
LOCAL
OSCILLATOR
-90°
-90°
PHASE
SHIFT
NETWORK
SIN(ωt)
SIN(ωt)
LP
(LOW PASS)
MIXER
MIXER
Σ
LP
(LOW PASS)
AUDIO
AMPLIFIER

Fig.8: the Weaver receiver configuration has some advantages over Hartley but many more mixers are required, so the resulting noise performance is less than ideal (in the analog domain, anyway).

may have built a Weaver receiver and find what their results were.

## The ultimate design

Having tried all the different architectures (apart from TRF) over a period of about six months, I decided that the SSB superhet design was the most practical approach for home construction. But before we get to the circuit, here is an explanation of two modulation techniques.

Amplitude modulation (AM) is where a 'carrier frequency' signal is multiplied by an audio frequency (AF) signal, as shown in Fig.9. We get a signal with components in three frequency ranges: the original carrier, plus two 'side-bands', being the sum and difference (see Fig.10).

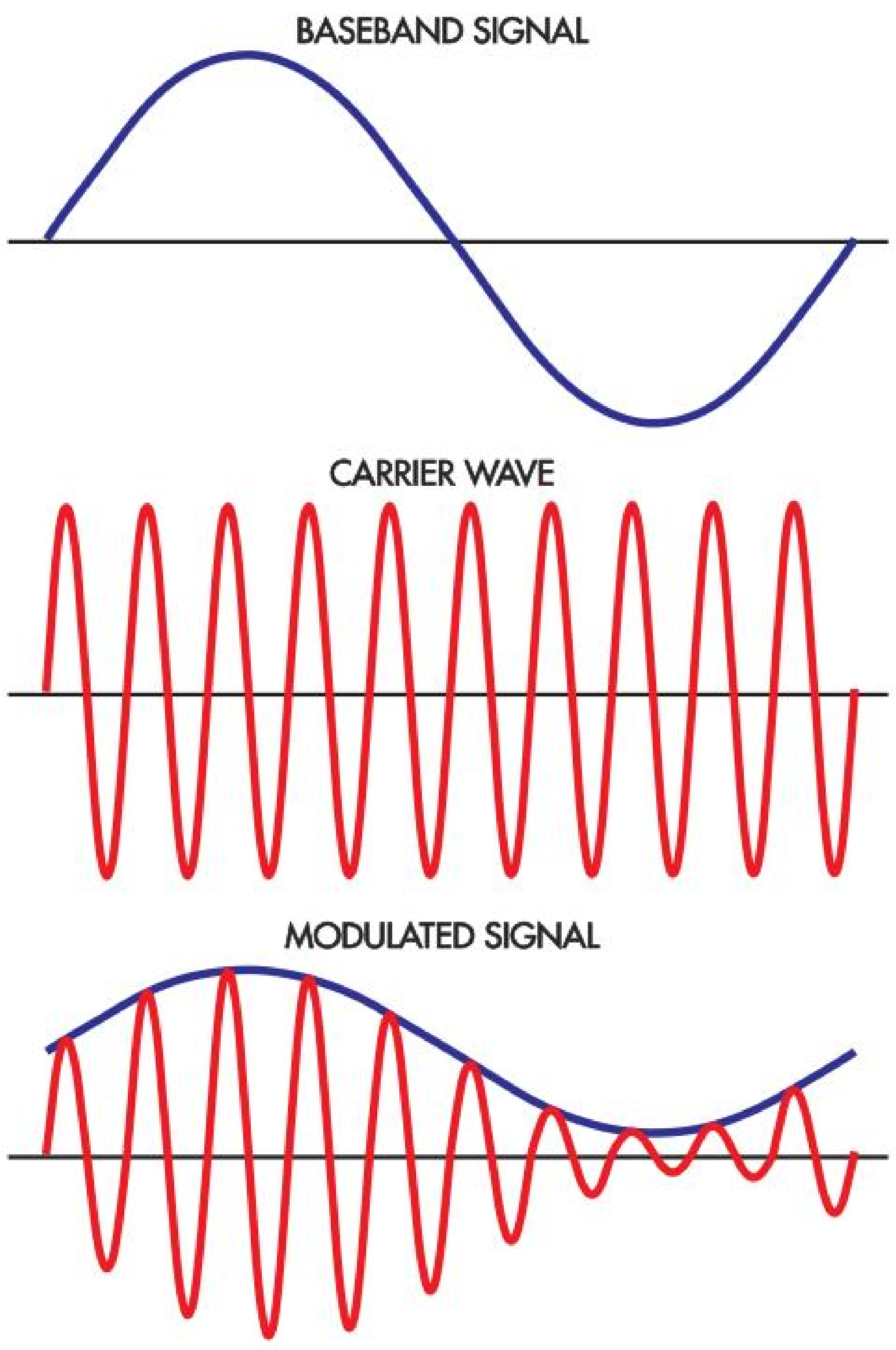


**Fig.9: the basic principle of amplitude modulation (AM). The high-frequency carrier amplitude varies with the instantaneous baseband signal amplitude.**

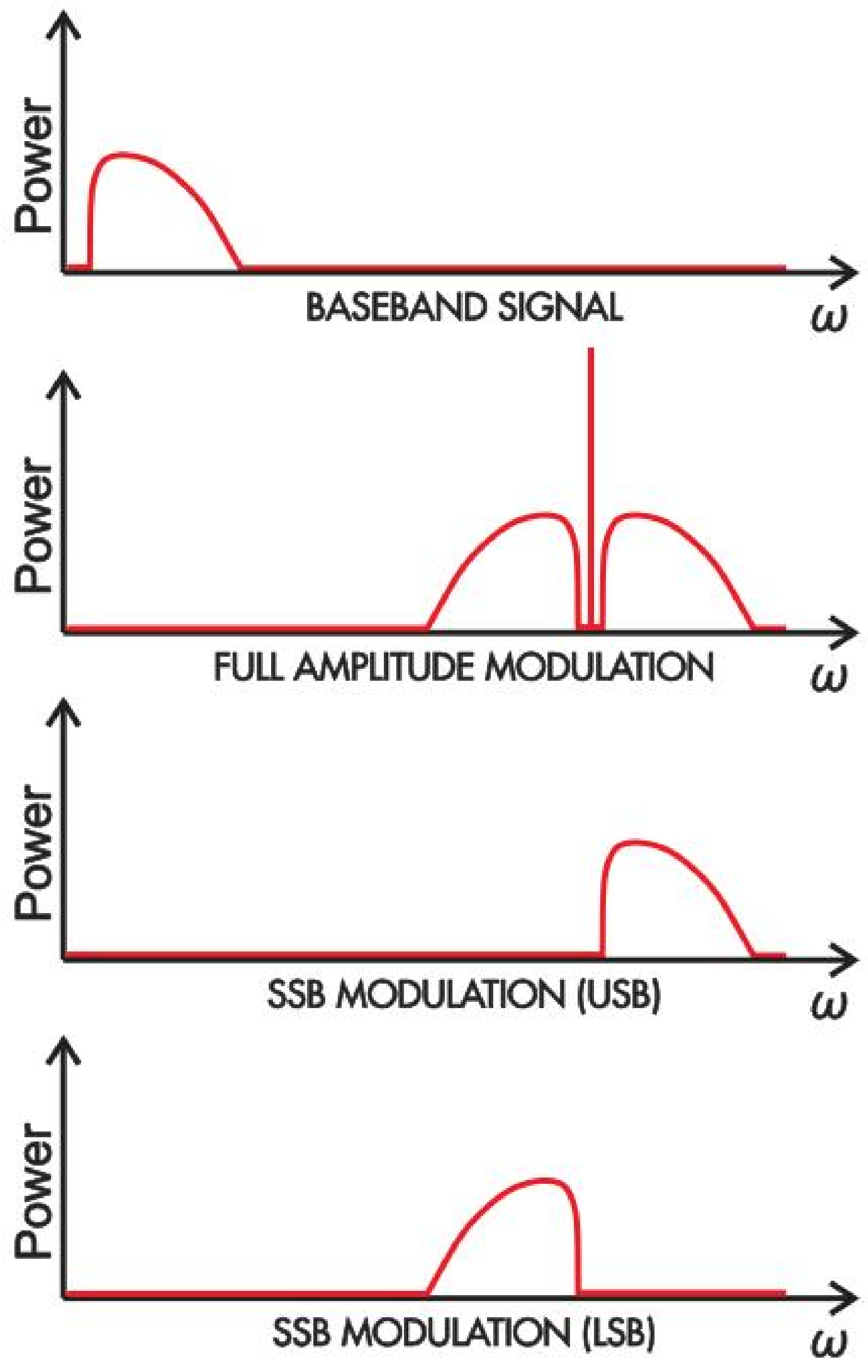


**Fig.10: the spectra of AM and SSB transmissions. The transmission power of SSB is about ¼ that of AM without significantly reducing the received signal strength.**

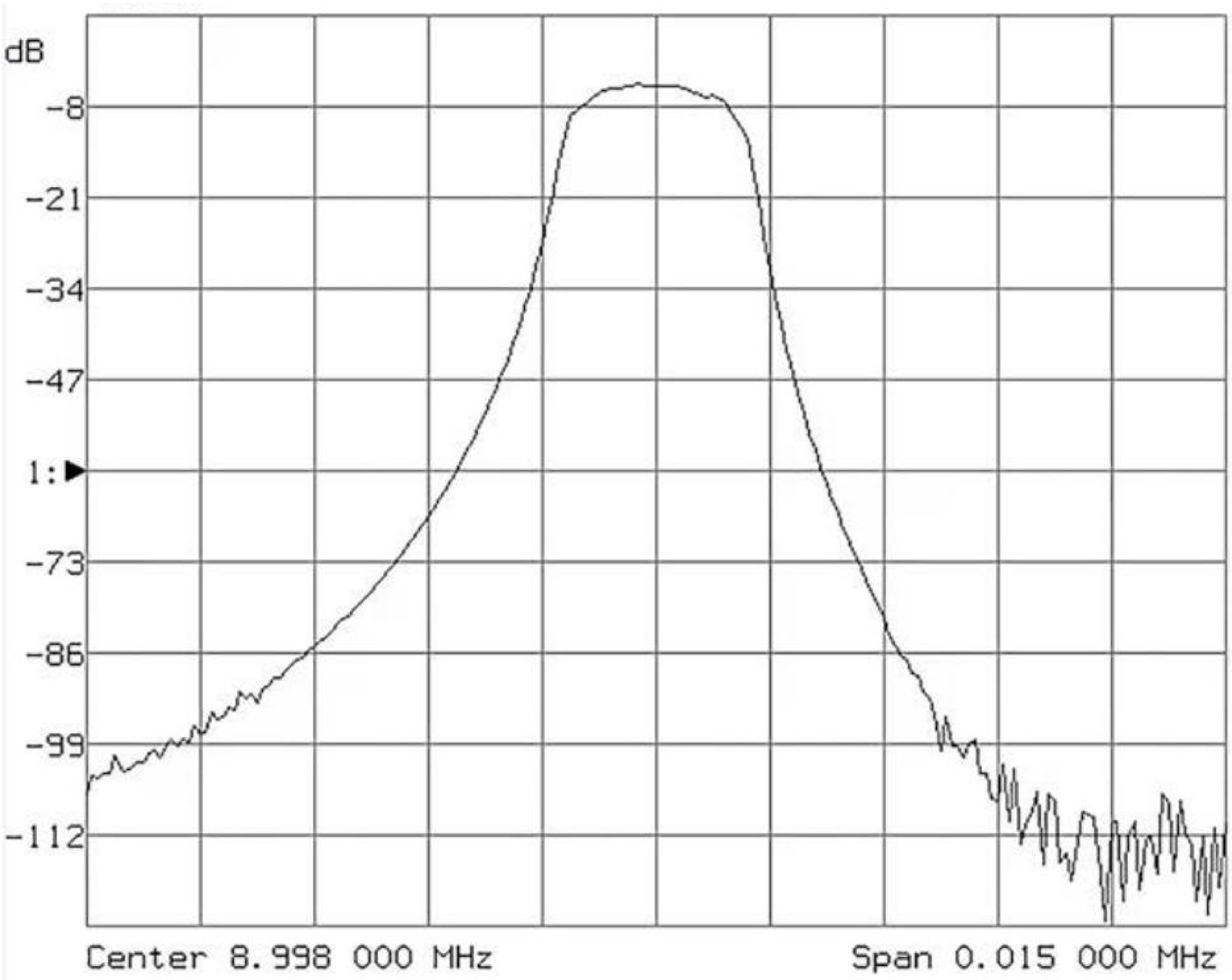


**Fig.11: the measured performance of the pre-built 9MHz crystal filter module. It combines a flat passband with very steep roll-offs on either side.**

To demodulate the AM signal, all that is needed is a diode and a low-pass filter to remove the RF component. This filter may be just a single resistor and capacitor.

While AM is easy to implement, is really quite wasteful. The carrier frequency carries no information at all, and the two side-bands at 100% modulation contain half the power of the carrier, with identical information.

This is where the single side-band (SSB) method of communication is far more efficient. We essentially get rid of the carrier and one of the side-bands. Instead of a bandwidth of twice the baseband, our filter needs only the baseband bandwidth. The spectrum for SSB modulation is shown at the bottom of Fig.10.

However, with SSB, the simple envelope detector will no longer work. To take an example, transmitting an SSB signal modulated at two frequencies, 1kHz and 2kHz, an envelope detector would give us a tone of 1kHz, being the difference between the two frequencies. For a more complex modulated signal, the output of the detector would be quite unintelligible.

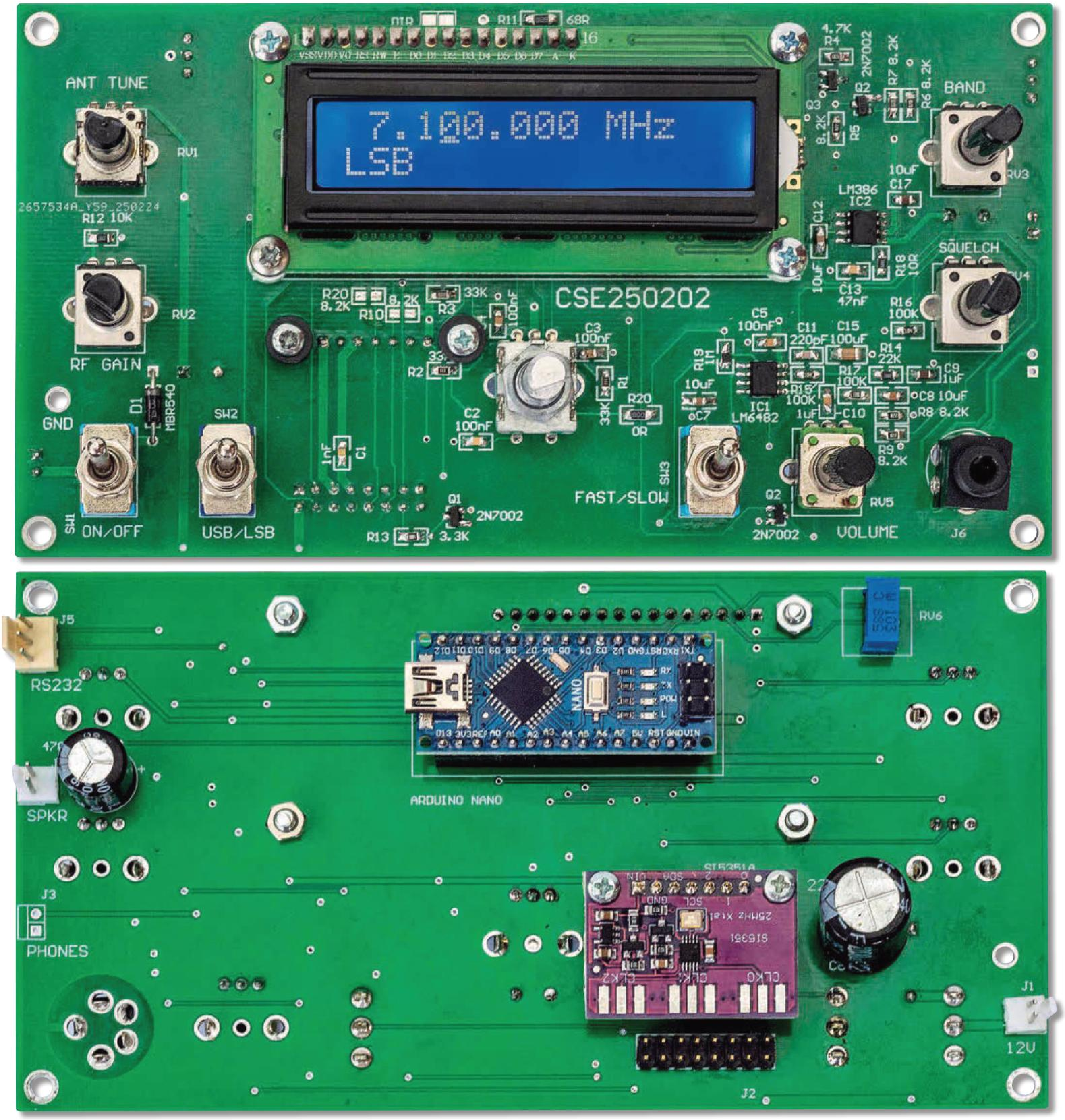
To recover the audio, we have to multiply the output of the IF amplifier with the signal from a beat frequency oscillator (BFO) with a second mixer. The BFO frequency is set to where the carrier frequency would otherwise be. This results in two signals in the output; one is the original baseband signal, plus another at twice the IF, which is easily removed by a low-pass filter.

The filtered signal can then be amplified by an audio amplifier to drive a speaker or headphones.

## Receiver design

The receiver presented here covers the frequency range of 3MHz to 30MHz, with an audio bandwidth limited to the frequencies used by human speech: 300Hz to 3kHz. This means that the IF filter bandwidth needs to be 2.7kHz (3kHz – 300Hz). This is best achieved by a multi-pole crystal filter at 9MHz.

This is quite a critical item in the design. You can build your own by buying a batch of 9MHz crystals and carefully selecting them for series


7.100.000 MHz
LSB
ANT TUNE
RV1
RF GAIN
RV2
BAND
RV3
SQUELCH
RV4
VOLUME
RV5
FAST/SLOW
SW3
ON/OFF
SW1
USB/LSB
SW2
GND
D1
MBR540
CSE250202
LM386
IC2
LM6482
IC1
2N7002
J6
RS232
J5
SPKR
PHONES
J3
ARDUINO NANO
SI5351A
RV6
12V
J1
J2

The front and rear sides of the control board. The five pots, three switches, rotary encoder, LCD screen and headphone socket form the user interface. On the rear of the control board are the Arduino Nano and clock generator modules, LCD adjustment trimpot, two electrolytic capacitors and some connectors. Note that these photos are shown enlarged for clarity.

and parallel resonant frequencies. But unless you have the equipment and patience to do this accurately, it is not worthwhile.

I bought 20 9MHz crystals for about £3, and by selection, managed a reasonable filter after much experimentation. But a complete six-pole filter module is available from AliExpress for about £13, with an excellent bandwidth, as is shown in Fig.11.

Next, let's look at how we deal with image frequencies. If the desired signal $f_S$ = 7MHz and the local oscillator $f_O$ = 16MHz, producing a 9MHz IF signal, a signal at 25MHz mixed with 16MHz will produce the same 9MHz IF. This is why we have an input tuned circuit.

How sharp does this filter have to be? Using a high-quality toroid, a loaded Q of 100 is typical. There are calculators on the internet that save us the trouble of laboriously working it out; with the above example, the unwanted 25MHz image signal will be attenuated by about 50dB.

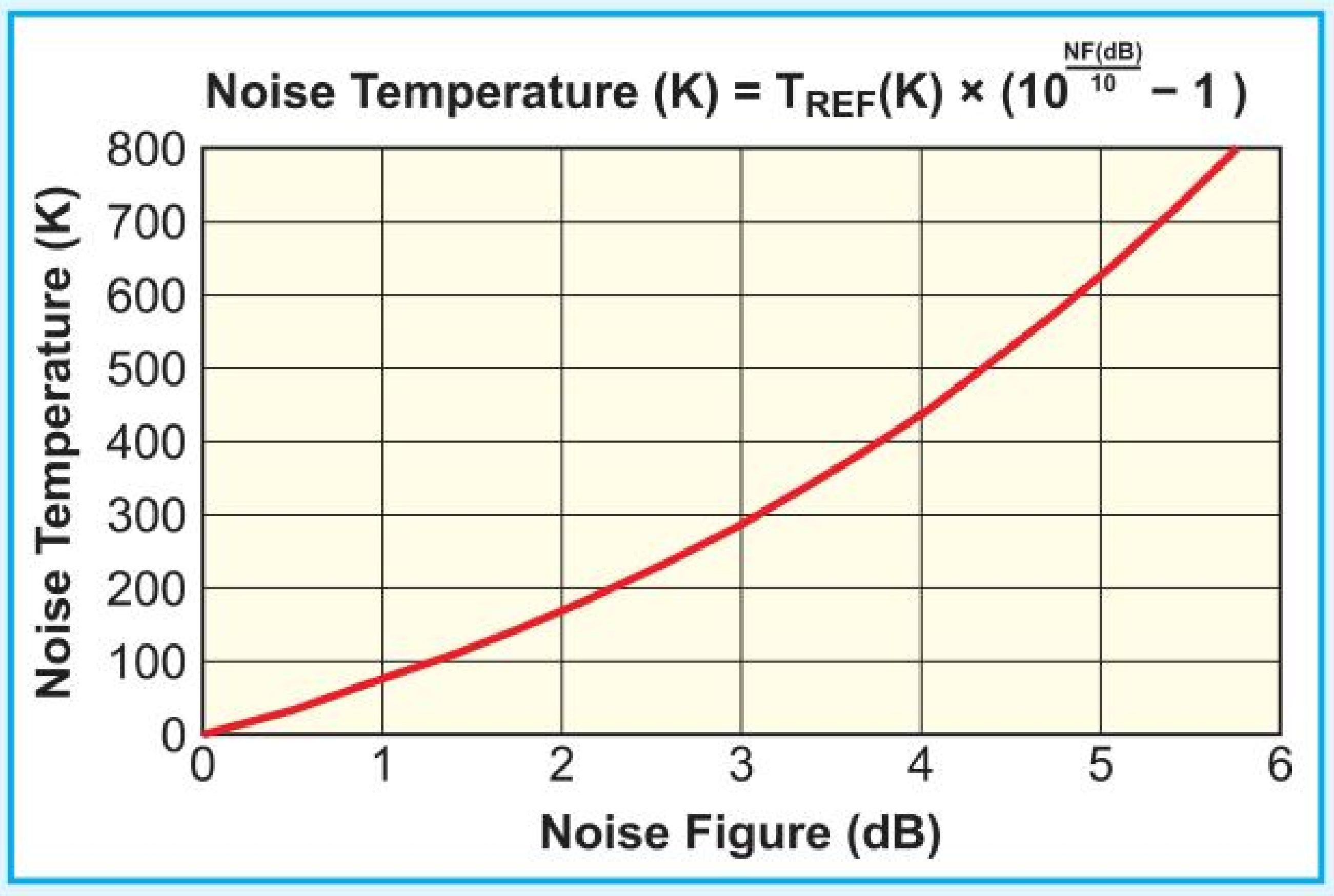
That is why a relatively high IF is

## What is a noise figure?

Every device generates broadband noise that will reduce the circuit's signal-to-noise ratio (SNR). The NF is the ratio of actual output noise to that which would remain if the device itself did not introduce noise, which is equivalent to the ratio of input SNR to output SNR.

There is another way of expressing the noise performance: the noise temperature, expressed in Kelvins as an equivalent temperature. It is not the physical temperature of a system, but a theoretical value that defines the temperature required to produce a specific amount of noise power.

The equivalence between noise temperature and noise figure is shown below. The reference temperature, $T_{REF}$, is generally 290K (16.85°C).



**The relationship between noise figure (NF) and noise temperature (in Kelvin). Note that it is not the actual temperature the part is operating at.**

desirable for higher-frequency signals, as the image frequency is well removed. If an IF of 455kHz were used, the standard for broadcast-band receivers, the image at 7.91MHz would only be 28dB down.

### Circuit details

Figs.12 & 13 show the full circuit of the receiver, which is split across two PCBs, and the circuits correspond to them. One is the control board, while the other is the RF board.

At the heart of the control board is the Arduino Nano module, which has the ATMega328 microcontroller. The display is the common 16×2 alphanumeric LCD module; the version with a blue backlight is the best choice. Potentiometer VR6 is the contrast adjustment for the LCD screen.

The variable frequency oscillator (VFO) and the beat frequency oscillator (BFO) signals are generated by an Si5351A clock-generator module (MOD2), controlled over an I²C serial bus (SDA/SCL). This module can generate three different frequencies as square waves with amplitudes of about 3V peak-to-peak. In this design, the outputs used are CLK0 and CLK2; the CLK1 output is not used.

The 8.2kΩ pull-up resistors for the SDA & SCL lines are shown greyed out in Fig.12 because they do not need to be fitted as the Si5351A module has onboard pull-up resistors.

A rotary encoder (RE1) is used for frequency tuning; the step size for each click can be varied using the integrated pushbutton switch. Pressing the switch cycles through steps of 10Hz, 100Hz, 1kHz, 10kHz, 100kHz and 1MHz. The two poles of the encoder, plus its integral switch, have 33kΩ pull-up resistors to give defined high/low levels and 100nF capacitors to ground for debouncing.

The I²C serial bus is used to control the input circuit tuning by selecting six capacitors in various combinations for approximate tracking with frequency. Fine potentiometer VR1 is used to change the voltage on a varicap diode to interpolate the approximate values and peak the input circuit to resonance (more on this later).

There is audio circuitry on this board as well. Op amp IC1b has a gain of about 5.5, and can drive headphones directly via a 3.5mm jack provided on the front panel. It has internal switching that disconnects the power amplifier driving the speaker when headphones are plugged in.

Despite the existence of numerous more modern power amplifier chips, I have used the venerable LM386 (in an SMD package) to drive the speaker. It requires few external components, is cheap and with a 12V supply will deliver over 2W to an 8Ω speaker

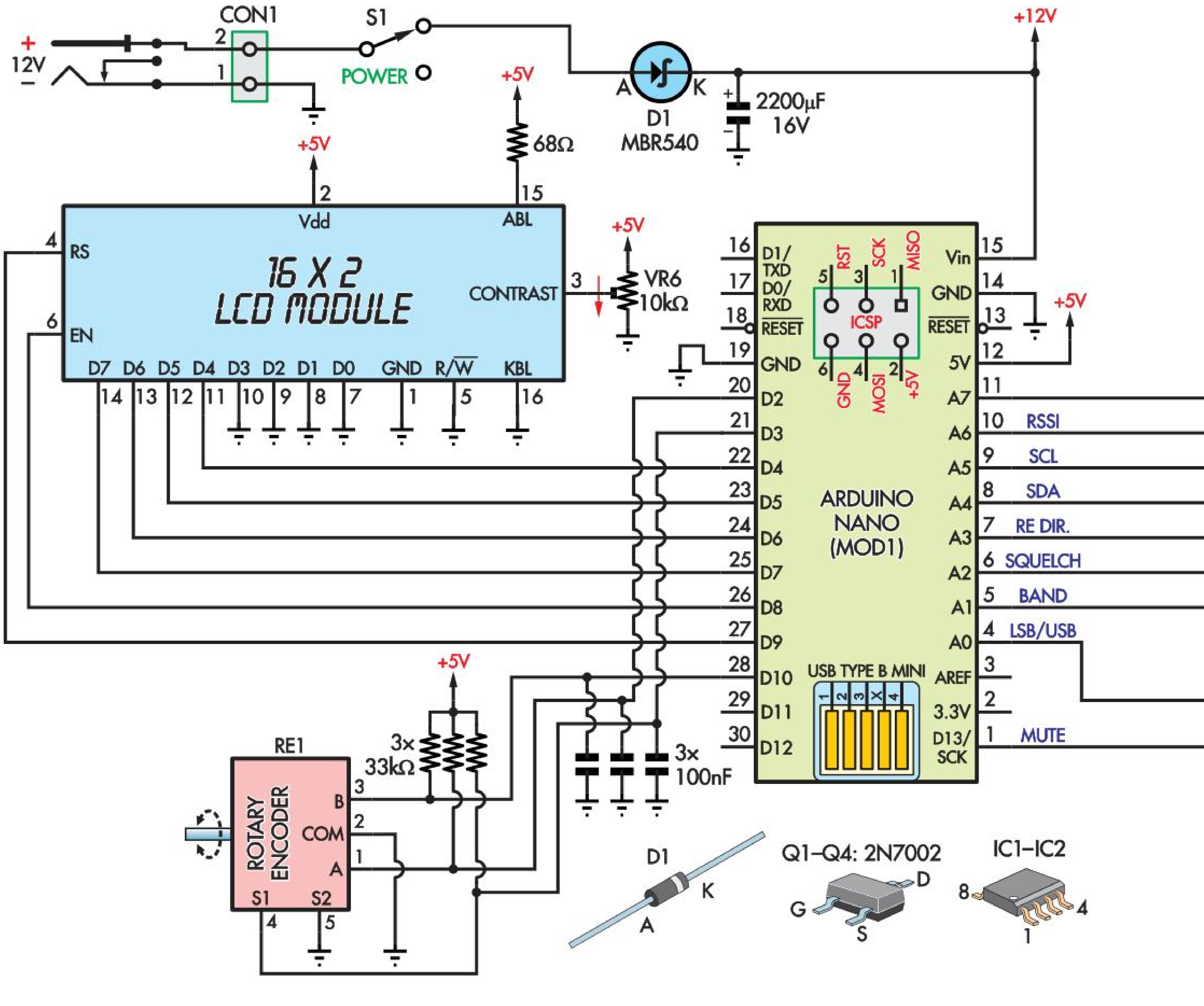

CON1
S1
+12V
+
12V
−
2
1
POWER
+5V
68Ω
A
K
D1
MBR540
2200µF
16V
+5V
2
15
Vdd
ABL
4
RS
6
EN
16 X 2
LCD MODULE
CONTRAST
3
+5V
VR6
10kΩ
D7 D6 D5 D4 D3 D2 D1 D0 GND R/W KBL
14 13 12 11 10 9 8 7 1 5 16
16 D1/TXD
17 D0/RXD
18 RESET
19 GND
20 D2
21 D3
22 D4
23 D5
24 D6
25 D7
26 D8
27 D9
28 D10
29 D11
30 D12
RST SCK MISO
ICSP
GND MOSI +5V
ARDUINO NANO (MOD1)
USB TYPE B MINI
Vin 15
GND 14
RESET 13
5V 12
A7 11
A6 10 RSSI
A5 9 SCL
A4 8 SDA
A3 7 RE DIR.
A2 6 SQUELCH
A1 5 BAND
A0 4 LSB/USB
AREF 3
3.3V 2
D13/SCK 1 MUTE
+5V
+5V
RE1
ROTARY ENCODER
3× 33kΩ
B 3
COM 2
A 1
S1 4
S2 5
3× 100nF
D1
K
A
Q1–Q4: 2N7002
D
G
S
IC1–IC2
8
4
1

**Fig.12: the control board circuit. Three main modules are used: the Arduino Nano 'brain', an Si5351 digital clock generator that produces the VFO and BFO oscillator signals and the 16×2 alphanumeric LCD module. The dual op amp provides the squelch function (IC1a) and audio gain for driving headphones (IC1b), while IC2 is the power amplifier that drives the speaker.**

# SSB Shortwave Receiver, part one

The control board and RF board are joined by a 16-wire flat cable between headers CON2. This supplies power to the RF board and carries signals to it as well, including the band change signal and the I²C bus (SDA & SCL).

Signals coming into the control board on CON2 include the recovered audio and RSSI (received signal strength indicator) voltage.

Potentiometer VR5 is the volume control, while VR2 is an RF gain control modifying the AGC (automatic gain control) voltage. Switch S3 is the SLOW/FAST AGC selection, adding a 10µF capacitor to the RSSI line for the SLOW AGC mode.

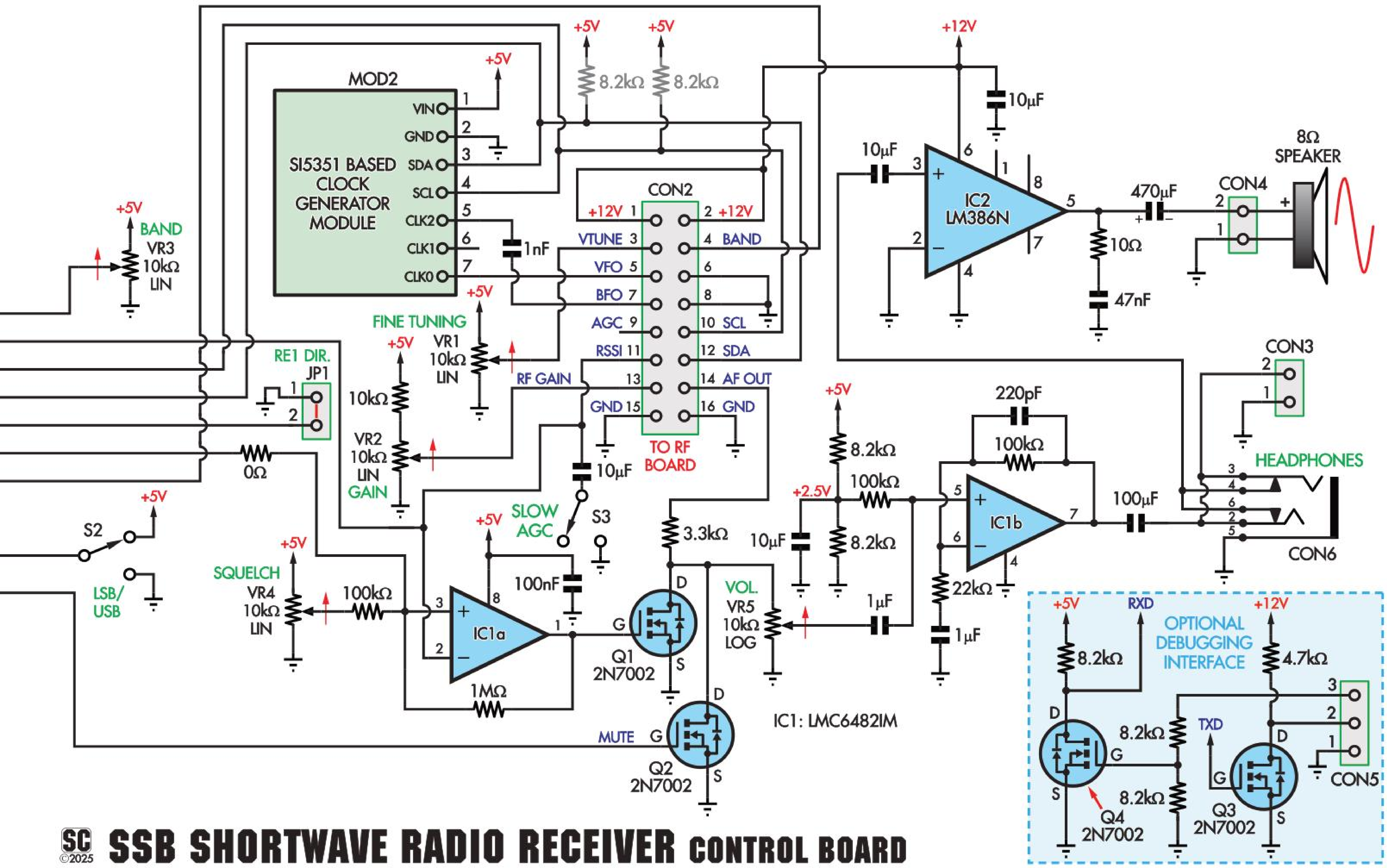
The squelch control is useful in eliminating background noise from weak input signals. It works by comparing the RSSI voltage level at the inverting input of IC1a (which is used as a voltage comparator) with a DC voltage derived from potentiometer VR4. When the RSSI level is low, Mosfet Q1 is switched on, shorting out the audio. The 1MΩ feedback resistor provides hysteresis.

SSB RECEIVER
ANT TUNE
RF GAIN
3.609.000 MHz
LSB
BAND
SQUELCH
USB
ON
LSB
FAST
AGC
SLO
VOLUME
HONES

**The controls are all labelled on the front panel PCB. The rear of the set only has the BNC antenna terminal, DC power connector and holes so that sound from the the internal speaker can escape.**

The mute function is provided by a second transistor, Q2, in parallel with Q1. I found that when the frequency was being changed, there was a loud annoying click in the audio. So during tuning, Q2 is switched on, also shorting out the audio.

DC power is via CON1 and diode D1 protects against the wrong supply polarity. The supply voltage can range from 9-12V DC, with a maximum current drain of about 250mA. An iron-core transformer based plugpack is preferable as it does not generate RF noise, but you can try a switching plugpack; some do have low noise.


MOD2
SI5351 BASED CLOCK GENERATOR MODULE
VIN 1
GND 2
SDA 3
SCL 4
CLK2 5
CLK1 6
CLK0 7
+5V
8.2kΩ
8.2kΩ
+12V
10µF
1nF
CON2
+12V 1
2 +12V
VTUNE 3
4 BAND
VFO 5
6
BFO 7
8
AGC 9
10 SCL
RSSI 11
12 SDA
13
14 AF OUT
GND 15
16 GND
TO RF BOARD
RF GAIN
BAND
VR3 10kΩ LIN
FINE TUNING
VR1 10kΩ LIN
RE1 DIR.
JP1
10kΩ
VR2 10kΩ LIN GAIN
0Ω
10µF
SLOW AGC
S3
S2
LSB/USB
SQUELCH
VR4 10kΩ LIN
100kΩ
IC1a
100nF
1MΩ
3.3kΩ
Q1 2N7002
Q2 2N7002
MUTE
D G S
VOL.
VR5 10kΩ LOG
IC1: LMC6482IM
+2.5V
8.2kΩ
100kΩ
10µF
220pF
100kΩ
IC1b
22kΩ
1µF
1µF
100µF
IC2 LM386N
10µF
470µF
10Ω
47nF
CON4
8Ω SPEAKER
CON3
HEADPHONES
CON6
OPTIONAL DEBUGGING INTERFACE
RXD
TXD
8.2kΩ
4.7kΩ
8.2kΩ
8.2kΩ
Q4 2N7002
Q3 2N7002
CON5
SC ©2025

**SSB SHORTWAVE RADIO RECEIVER CONTROL BOARD**

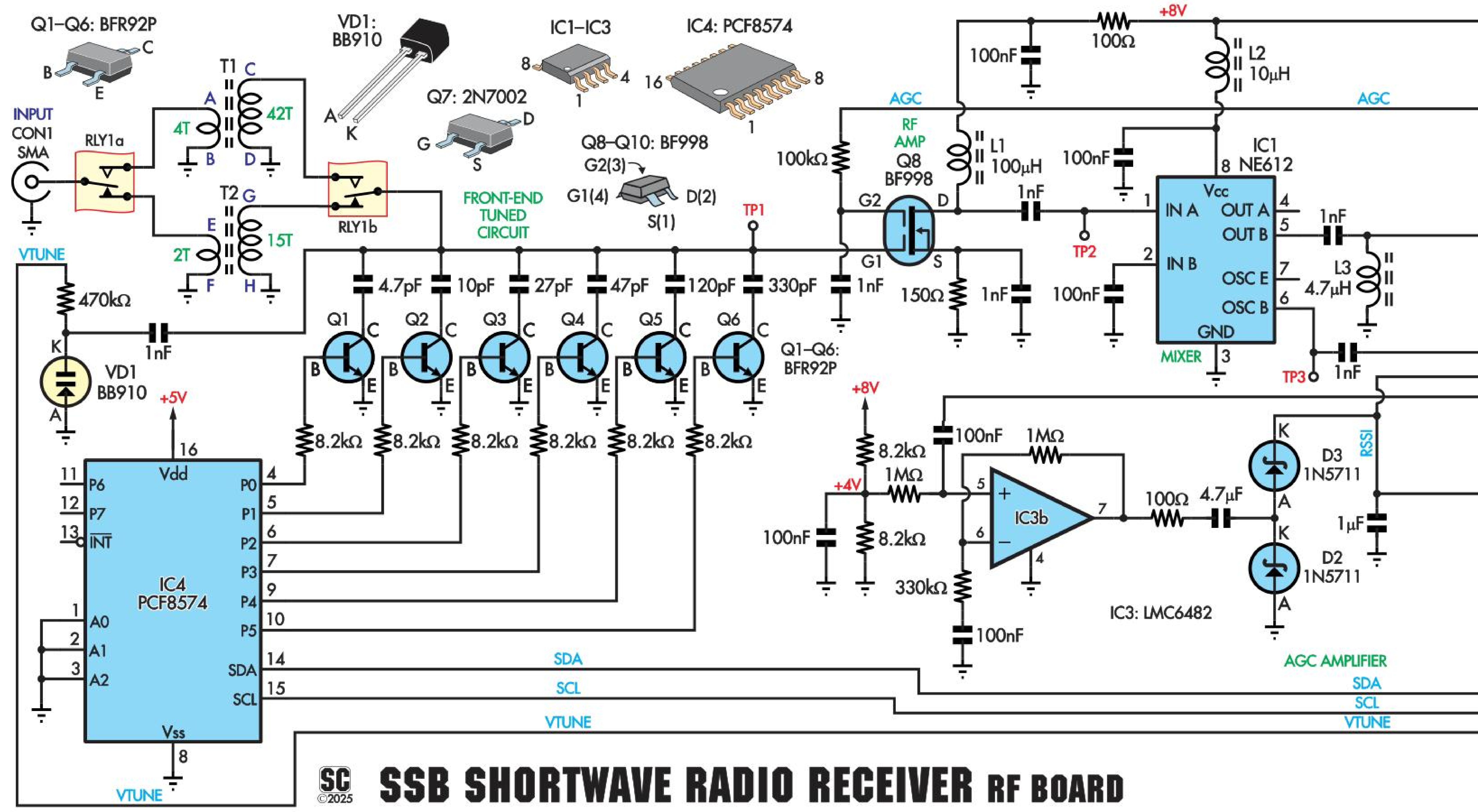

Q1–Q6: BFR92P
B
C
E
INPUT
CON1
SMA
RLY1a
T1
A
C
4T
42T
B
D
T2
G
E
2T
15T
F
H
VD1: BB910
A
K
RLY1b
IC1–IC3
8
4
1
IC4: PCF8574
16
8
1
Q7: 2N7002
G
D
S
Q8–Q10: BF998
G2(3)
G1(4)
D(2)
S(1)
FRONT-END TUNED CIRCUIT
TP1
VTUNE
470kΩ
K
VD1
BB910
A
1nF
4.7pF
10pF
27pF
47pF
120pF
330pF
Q1
Q2
Q3
Q4
Q5
Q6
Q1–Q6: BFR92P
8.2kΩ
+5V
16
Vdd
IC4
PCF8574
P6
P7
INT
A0
A1
A2
P0
P1
P2
P3
P4
P5
SDA
SCL
Vss
11
12
13
1
2
3
4
5
6
7
9
10
14
15
8
SDA
SCL
VTUNE
VTUNE
100nF
100Ω
+8V
L2
10μH
AGC
AGC
100kΩ
RF AMP
Q8
BF998
L1
100μH
100nF
IC1
NE612
G2
D
G1
S
1nF
150Ω
1nF
TP2
100nF
Vcc
IN A
IN B
OUT A
OUT B
OSC E
OSC B
GND
MIXER
L3
4.7μH
TP3
1nF
+8V
8.2kΩ
+4V
1MΩ
100nF
8.2kΩ
100nF
1MΩ
5
6
7
4
IC3b
330kΩ
100nF
100Ω
4.7μF
IC3: LMC6482
K
D3
1N5711
A
RSSI
1μF
K
D2
1N5711
A
AGC AMPLIFIER
SDA
SCL
VTUNE
SC ©2025
SSB SHORTWAVE RADIO RECEIVER RF BOARD

**Fig.13: this RF board circuit connects to the control board circuit (Fig.12) via CON2. Q1-Q6 and VD1 tune the incoming signal while T1 (3-10MHz) or T2 (10-30MHz) are selected by RLY1 for band switching. Q8 is the RF gain stage; IC1 is the superhet mixer; Q9 is the first IF gain stage; Q10 is the second IF gain stage; IC2 is the BFO mixer; and dual op amp IC3 is the RSSI/AGC signal amplifier.**

Because the voltage regulation of iron-core plugpacks is poor, a 12V one may put out a voltage that is too high with a light load, so choose one rated at 9V DC and 500mA. Depending on the Arduino Nano, the voltage regulator may not tolerate an input voltage much greater than 12V, so be careful with the choice. You can easily blow up a Nano with excessive input voltage (trust me, I have!).

The filtering on this type of plugpack may leave too much 100Hz ripple, which would be heard as hum in the output. That's why there is a 2200μF electrolytic capacitor after D1. When S1 is switched on, there is a very high inrush current to charge this capacitor, hence a fairly high-current schottky diode is used for D1.

The ideal supply would be a ~12V battery; three 18650 cells in series give just over 11V fully charged. The background noise using a battery is significantly lower than either type of mains-powered supply. A suitable battery holder can be squeezed into the case, although taking out the cells to charge them requires removing a bracket. Still, you could integrate a charging socket.

Most constructors will not need the optional serial debugging interface provided by Mosfets Q3 & Q4; they offer a bidirectional RS-232 compatible serial stream at header CON5. Those components can be left off if not needed. You can also connect a TTL USB/serial adaptor directly to the TXD & RXD pins of MOD1.

## RF module circuit

As shown in Fig.13, the signal from the antenna goes to two tuned toroidal transformers selected by relay RLY1. A high Q is desirable in these transformers for maximal rejection of unwanted frequencies. The toroids are Micrometals T37-17 types with an unloaded Q in excess of 200 at most frequencies. With a 50Ω source on the primary winding, the loaded Q will be about 100.

Transformer T1 covers the range of 3-10MHz and has a secondary inductance of 7.4μH (42 turns). The antenna winding is four turns at the 'cold end' of the toroid. This needs a capacitance range from 34pF at 10MHz to 380pF at 3MHz for tuning.

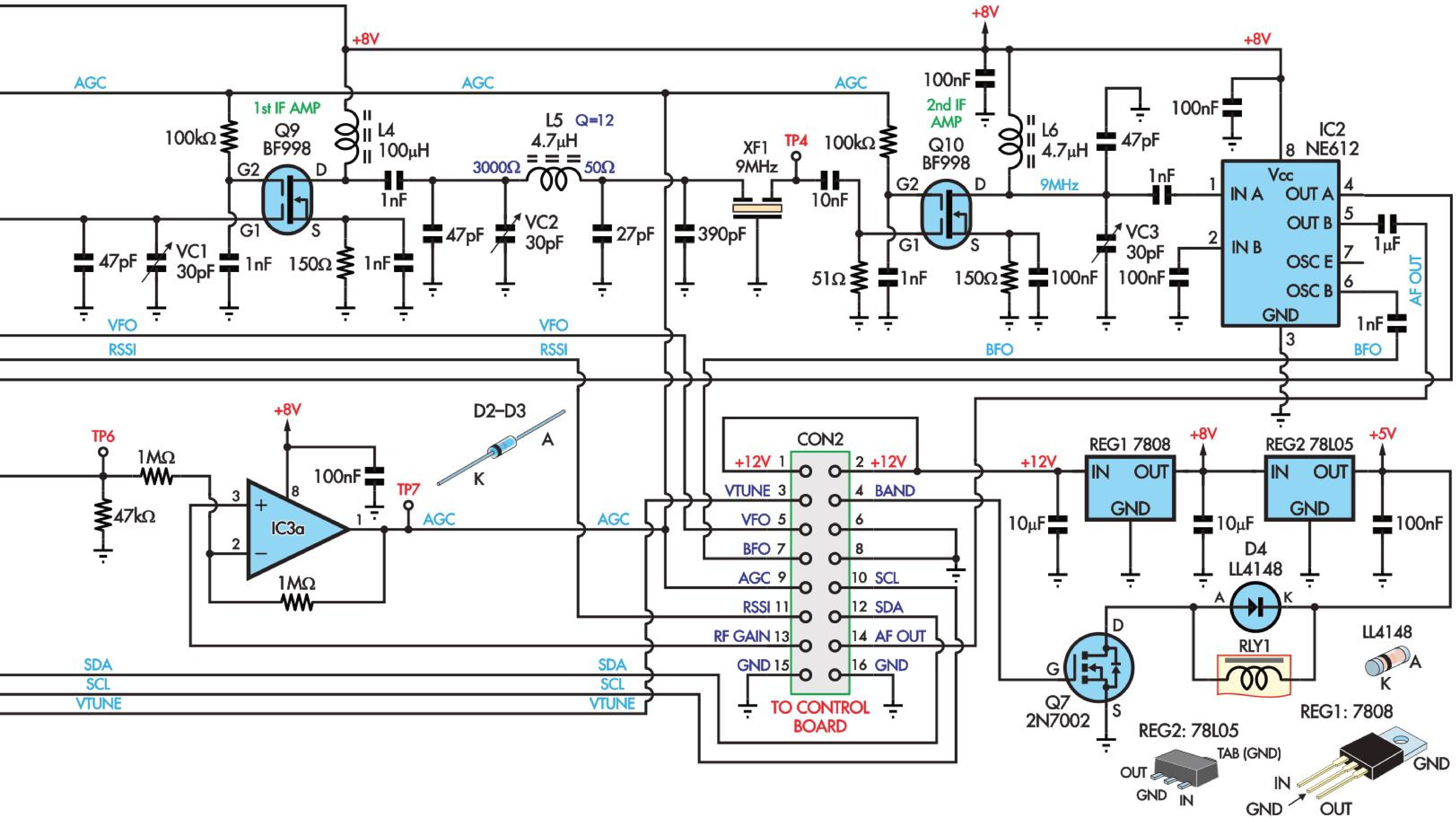
Back in the days when valves were used, this capacitance would be part of a two- or three-gang variable capacitor. These days, such capacitors are relatively rare, expensive and too large. Instead, I used six fixed capacitors selected by the PCF8574 I$^2$C extender (IC4) driving six NPN RF transistors.

A BB910 varicap diode (VD1) adds to the tuning capacitance as a fine adjustment to interpolate between the fixed values. The capacitance range of the varicap is from 40pF at 0.5V down to 8pF at 9V (a varicap diode is used in reverse bias, with the voltage across it affecting its capacitance).

For the range from 10.1MHz to 30MHz, we switch in transformer T2, with an inductance of 1.1μH (15 turns). Its antenna winding is two turns.

Q8 is a low-noise amplifier based on a BF998 dual gate Mosfet. While technically obsolete, this is easily obtainable from many sources. Rather than another tuned circuit in the drain, I have just used a 100μH inductor, which has a reasonably high impedance over the entire 3-30 MHz range. The gain is about 20dB and, while the noise figure (NF) is not given below 30MHz, at 800MHz it is 1dB.

The first mixer (IC1) is an NE612 (or SA612) IC. This has a gain of about


+8V
AGC
1st IF AMP
Q9
BF998
100kΩ
L4
100μH
G2
D
G1
S
47pF
VC1
30pF
1nF
150Ω
1nF
1nF
47pF
L5 Q=12
4.7μH
3000Ω
50Ω
VC2
30pF
27pF
390pF
XF1
9MHz
TP4
10nF
100kΩ
51Ω
1nF
2nd IF AMP
Q10
BF998
100nF
L6
4.7μH
9MHz
150Ω
100nF
47pF
VC3
30pF
100nF
100nF
1nF
IC2
NE612
Vcc
IN A
IN B
OUT A
OUT B
OSC E
OSC B
GND
1μF
AF OUT
1nF
BFO
VFO
RSSI
TP6
1MΩ
47kΩ
IC3a
100nF
TP7
1MΩ
D2–D3
A
K
CON2
+12V 1
2 +12V
VTUNE 3
4 BAND
VFO 5
6
BFO 7
8
AGC 9
10 SCL
RSSI 11
12 SDA
RF GAIN 13
14 AF OUT
GND 15
16 GND
TO CONTROL BOARD
SDA
SCL
VTUNE
+12V
REG1 7808
IN OUT GND
10μF
10μF
REG2 78L05
+5V
100nF
D4
LL4148
RLY1
Q7
2N7002
G D S
LL4148
REG2: 78L05
TAB (GND)
OUT GND IN
REG1: 7808
IN GND OUT
GND

17dB and a noise figure of 5dB. The NF of a multi-stage amplifier can be calculated as:

$$NF = NF1 + (NF2 - 1) \div G1 + (NF3 - 1) \div (G1 \times G2) + \ldots$$

Here, NF is the total noise figure, while NF1, NF2... are the noise figures of subsequent stages, and G1, G2... the gains of the stages. Thus, with a reasonably high gain in the first stage, the overall noise figure is degraded only slightly by the following stages.

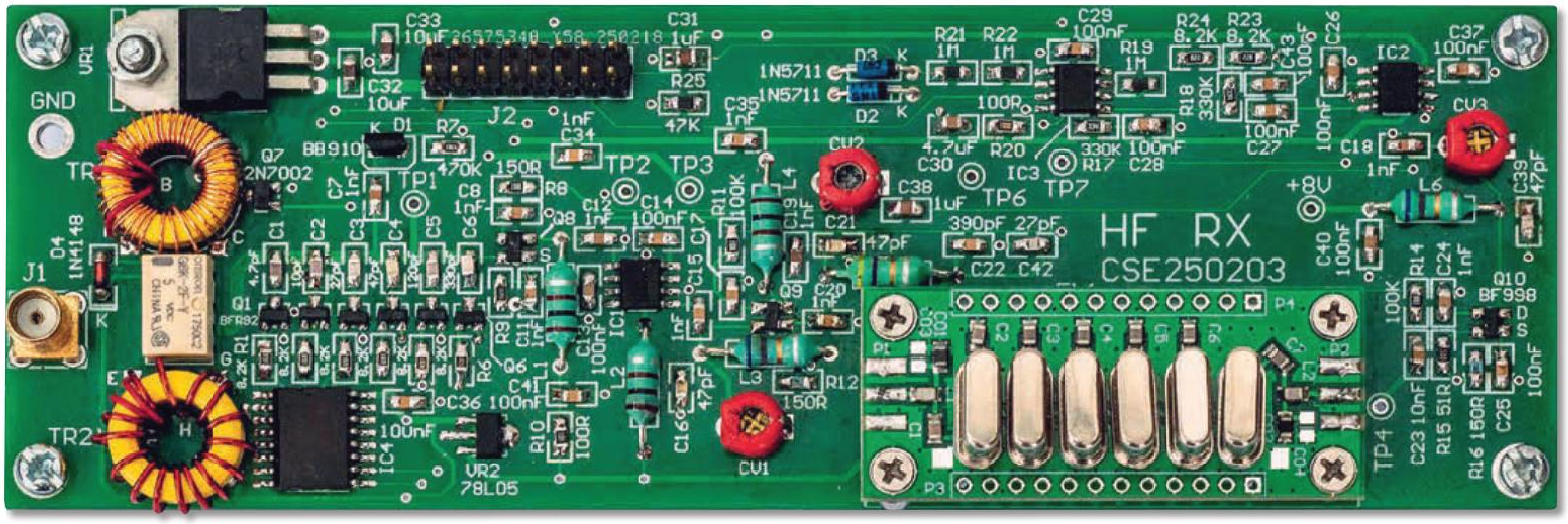
The final noise figure of our circuit is about 1.5dB. That is more than adequate given the amount of ambient noise in the HF band.

Another BF998 (Q9) follows the first mixer, providing another 20dB of gain. The 9MHz crystal filter follows, which has 50Ω input and output impedances. This is matched to the preceding BF998 amplifier by a pi network with a 3000:50Ω ratio. The filter introduces a loss of about 5dB. Another BF998 (Q10) is the second 9MHz IF amplifier after the crystal filter.

A second NE612 (IC2) is used for the second mixer, with the ~9MHz BFO. There are two outputs available on the chip. One output is connected to op amp IC1b, which has a voltage gain of about 46 times. This is the AGC amplifier. Schottky diodes D2 and D3 rectify this voltage and charge a 1μF capacitor. This voltage is applied to the inverting input of IC3a.

With no signal, this voltage is close to zero. The non-inverting input has a voltage from the RF gain control potentiometer on the front panel, and with the resistor values used, the output of IC3a is a maximum of about 4.5V. This is applied to the second gate of the three BF998 transistors for maximum gain.

As the RSSI voltage rises, the AGC voltage drops, going down to zero for very strong signals for minimum gain.


HF RX
CSE250203
GND
J1
TR1
TR2
CU1
CU2
CU3
IC1
IC2
IC3
Q7
2N7002
Q10
BF998
VR2
78L05
TP1
TP2
TP3
TP4
TP6
TP7
+8V

**The assembled RF board - toroidal transformers T1 & T2 are on the left, while the crystal filter module is at lower right. Note that this photo is shown enlarged for clarity.**

The maximum supply voltage for the NE612 mixers is 9V, so 8V is provided by a 7808 regulator. With a 9V main supply voltage, this will drop to about 7.5V, which is quite adequate. The PCF8574 I$^2$C I/O expander driving Q1–Q6 is the only chip that needs a 5V supply, which is provided by an SMD 78L05 regulator (REG2).

### Obtaining the components

I have been careful in choosing components that are readily available from many suppliers. Virtually all can be purchased from AliExpress (**www.aliexpress.com**) at quite low prices. For example, the modules on the control board are an Arduino Nano, 16×2 alphanumeric LCD and Si5351a, which can be bought for a grand total of about £5 plus shipping (a couple more pounds).

Although some components are classed as 'obsolete', they are all still readily available. That includes the BF998 dual gate Mosfets and NE612 ICs.

The LMC6482 op amp was chosen as it has a very high input impedance, an adequate GBW (gain bandwidth) of 1.5MHz but, most importantly, it is a rail-to-rail input/output type and can be used with a single supply voltage of up to 16V.

While the BF998 is easily obtainable, be careful not to use the BF998R, which has a mirror image pinout (mounting it upside-down is not easy!).

The most expensive component is the 9MHz crystal filter module, costing about £13. As I mentioned earlier, it's cheaper to build your own, but it requires the right equipment and is quite a bit of effort.

The other expensive item is the case. The metal case that I have specified is available for about £20. It comes with steel front and back panels. The front panel is replaced by a 1.6mm-thick black circuit board that has all the necessary holes and cutouts. Thus, you only need to drill holes in the back panel for the power, antenna connection, and loudspeaker (if fitted).

### Next month

We've covered quite a lot in this article, so the construction details will be in a follow-up article next month. It will also cover programming the Arduino Nano, preparing the case, plus calibrating and aligning the Receiver. **PE**

## Parts List – SSB Shortwave Receiver

1 180 × 130 × 110mm blue vented steel project box with feet
**[AliExpress 1005009324189095]**
1 assembled control board (see below)
1 assembled RF board (see below)
1 front-panel PCB coded CSE250204, 165.5 × 97mm, with black solder mask
1 panel-mount DC barrel socket, diameters to match plugpack
1 12V 500mA+ plugpack
1 50 × 90mm 8Ω loudspeaker (SPK1) **[AliExpress 32947164732]**
5 13mm diameter universal knobs **[AliExpress 1005006143033779]**
1 25mm diameter universal machined aluminium knob
1 10cm male SMA to female BNC panel-mount connector cable
**[AliExpress 1005007121661805 select "BNC F Nut to SMA M", "4inch (10CM)"]**
2 16-way IDC connectors
2 2-way 2.54mm pitch polarised header plugs with matching pins
1 20cm length of 16-wire ribbon cable
4 M4 × 10mm panhead machine screws, nuts & washers (for mounting SPK1)
4 M3 × 15mm tapped spacers
4 M3 × 10mm tapped spacers
12 M3 × 6mm panhead machine screws
4 M3 × 6mm black panhead machine screws

**Control board**
1 double-sided PCB coded CSE250202, 150 × 79.5mm
1 Arduino Nano programmed with CSE25020A.HEX (MOD1)
1 Si5351A clock generator module (MOD2) **[AliExpress, eBay etc]**
1 16×2 alphanumeric blue backlit LCD module (LCD1)
1 pulse-type PCB-mounting rotary encoder with integral switch and 20mm shaft (RE1)
4 10kΩ 9mm vertical PCB-mounting linear potentiometers with 20mm shafts (VR1-VR4)
1 10kΩ 9mm vertical PCB-mounting log potentiometer with 20mm shafts (VR5)
1 10kΩ multi-turn trimpot (VR6)
3 miniature SPDT toggle switches with solder tags (S1-S3)
3 2-pin polarised headers, 2.54mm pitch (CON1, CON3, CON4)
1 8×2-pin header, 2.54mm pitch (CON2)
1 PJ-341 3.5mm vertical PCB-mounting jack socket (CON6) **[AliExpress]**
2 15-pin female headers, 2.54mm pitch (for MOD1)
1 7-pin header, 2.54mm pitch (for MOD2)
1 16-pin header, 2.54mm pitch (for LCD1)
4 5mm-long untapped spacers, 3mm inner diameter
4 M3 × 12mm panhead machine screws and matching nuts
2 M2 or M2.5 × 11mm tapped spacers
4 M2 or M2.5 × 6mm panhead machine screws

**Semiconductors**
1 LMC6482IM dual CMOS-input op amp, SOIC-8 (IC1)
1 LM386M audio amplifier, SOIC-8 (IC2)
2 2N7002 N-channel Mosfets, SOT-23 (Q1, Q2)
1 MBR540 40V 5A axial schottky diode (D1)

**Capacitors** *(all SMD M2012/0805 size 50V X7R unless noted)*
1 2200µF 16V through-hole electrolytic
1 470µF 16V through-hole electrolytic
1 100µF 6.3V M3216/1206 size
4 10µF 25V X5R/X7R
2 1µF
4 100nF
1 47nF
1 1nF NP0/C0G
1 220pF NP0/C0G

**Resistors** *(all SMD M2012/0805 size 1% unless noted)*
1 1MΩ
3 100kΩ
3 33kΩ
1 22kΩ
1 10kΩ
2 8.2kΩ
1 3.3kΩ
1 68Ω M3216/1206 size
1 10Ω
1 0Ω M3216/1206 size

**Additional parts for optional debugging interface**
1 3-pin polarised header, 2.54mm pitch (CON5)
2 2N7002 N-channel Mosfets, SOT-23 (Q3, Q4)
3 8.2kΩ SMD M2012/0805 size 1% resistors
1 4.7kΩ SMD M2012/0805 size 1% resistor

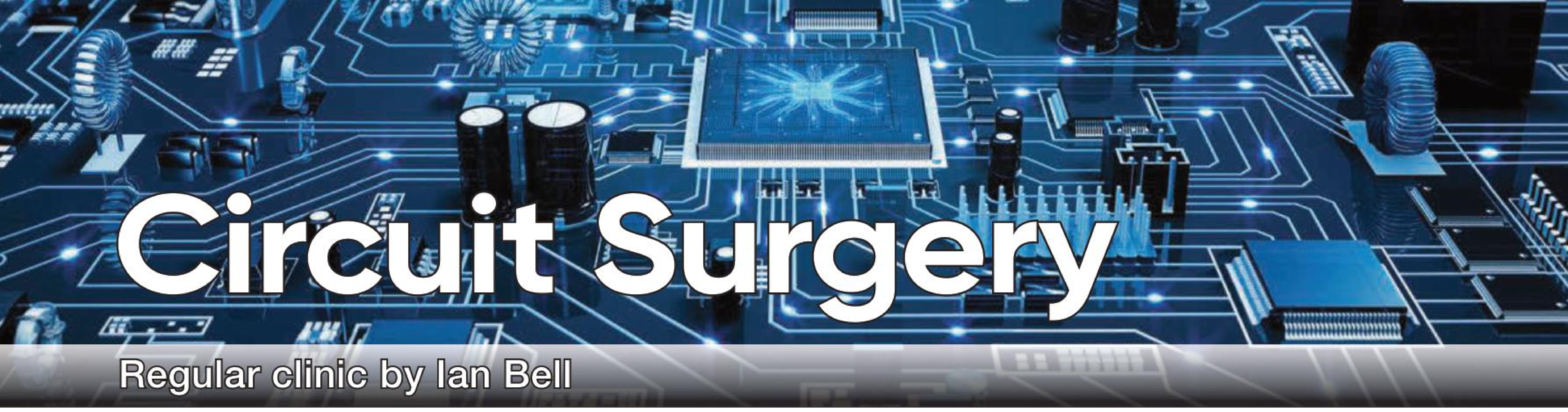
**RF board**
1 double-sided PCB coded CSE250203, 152 × 50mm
1 9MHz/600Hz crystal filter module (XF1) **[AliExpress 1005007201667282]**
2 100μH axial moulded inductors (L1, L4)
1 10μH axial moulded inductor (L2)
3 4.7μH axial moulded inductors (L3, L5, L6)
2 Micrometals Amidon T50-6 12.8mm toroidal cores (T1, T2) **[AliExpress]**
1 80cm length of 0.35mm diameter enamelled copper wire (T1)
1 30cm length of 0.6mm diameter enamelled copper wire (T2)
3 red 5-30pF trimmer capacitors (VC1-VC3)
1 vertical SMA connector, female, standard polarity (CON1)
1 8×2-pin header, 2.54mm pitch (CON2)
1 HFD4/5 or G6K-2F-Y 5V DC coil DIP DPDT signal relay (RLY1)
4 5mm-long untapped spacers, 3mm inner diameter
4 M3 × 10mm tapped spacers
1 M3 × 16mm panhead machine screw and matching nut (for REG1)
8 M3 × 6mm panhead machine screws
4 M2 or M2.5 × 12mm panhead machine screws and hex nuts

**Semiconductors**
2 NE612 oscillator/mixers, SOIC-8 (IC1, IC2)
1 LMC6482IM dual CMOS-input op amp, SOIC-8 (IC3)
1 PCF8574 $I^2C$ I/O expander, wide SOIC-16 (IC4)
1 7808 8V 1A linear regulator, TO-220 (REG1)
1 78L05 5V 100mA regulator, SOT-89 (REG2)
6 BFR92P low-noise RF NPN transistors, SOT-23 (Q1-Q6)
1 2N7002 N-channel Mosfet, SOT-23 (Q7)
3 BF998 dual-gate Mosfets, SOT-143 (Q8-Q10)
1 BB910 VHF varicap diode (VD1)
2 1N5711 axial schottky diodes (D2, D3)
1 LL4148 75V 200mA signal diode, SOD-80 (D4)

**Capacitors** *(all SMD M2012/0805 size 50V C0G/NP0 unless noted)*
2 10μF 25V X5R/X7R
1 4.7μF 25V X7R
2 1μF X7R
12 100nF X7R
1 10nF X7R
12 1nF
1 390pF
1 330pF
1 120pF
4 47pF
2 27pF
1 10pF
1 4.7pF

**Resistors** *(all SMD M2012/0805 size 1% unless noted)*
4 1MΩ
1 470kΩ
1 330kΩ
3 100kΩ
1 47kΩ
8 8.2kΩ
3 150Ω
2 100Ω
1 51Ω

PoLabs
For more information or software download please visit
www.poscope.com/epe
PoKeys
Connect, Control
- USB
- Ethernet
- Web server
- Modbus
- CNC (Mach3/4)
- IO
- PWM
- Encoders
- LCD
- Analog inputs
- Compact PLC
Stepper motor drivers
Drive
- up to 256 microsteps
- 50 V / 6 A
- USB configuration
- Isolated
- up to 32 microsteps
- 30 V / 2.5 A
PoScope Mega1+
PoScope Mega50
Measure
- up to 50MS/s
- resolution up to 12bit
- Lowest power consumption
- Smallest and lightest
- 7 in 1: Oscilloscope, FFT, X/Y, Recorder, Logic Analyzer, Protocol decoder, Signal generator



# Circuit Surgery

### Regular clinic by Ian Bell

## Measuring circuit frequency response using a PC sound card, part 8: more on op amp compensation

**In recent articles, we have** been looking at how to measure frequency response. This was motivated by the need to measure the response of a practical implementation of an example digital filter (from our series on DSP) using the sound card (audio interface) of a PC or laptop, together with software for this purpose.

As part of this topic, we have been discussing the signal conditioning that might be needed to interface the circuit under test to the sound card's line inputs and outputs. Most recently, we investigated op amp feedback, frequency response and stability.

Last month, we took a detailed look at the effect of capacitive loading on op amp amplifiers and its potential to cause instability (unwanted oscillation). This month we will consider various techniques for compensating op amp based amplifiers, ie, techniques to mitigate the instability problems.

### A recap on compensation

Several gain values are used when discussing op amps with feedback. Without feedback, an op amp has open-loop gain ($A_O$). With feedback, the circuit has a closed-loop gain, $A_C$. The value $-\beta A_O$ is the loop gain, which is the gain around the closed feedback loop, with a fraction ($\beta$) of the output fed back.

We often mention poles and zeros in the context of circuit stability. These are mathematical terms, but we can use the concepts in a practical sense to describe frequency responses.

Above a pole, the amplifier's gain decreases at a rate of 20dB per decade more than below it, and the phase shift becomes 90° more negative (assuming there are no other poles or zeros at similar frequencies). Above a zero, the gain increases at a rate of 20dB per decade more than below it, and the phase shift becomes 90° more positive.

As stated last month, a higher closed-loop gain means a smaller feedback fraction, so higher-gain circuits are generally more stable, with other aspects (such as the specific op amp used, loading and stray capacitance) being equal. Therefore, unity-gain buffers are the most challenging of the basic op amp circuits to achieve stability with, so they will be the focus of our examples.

All op amps would produce circuits that are unstable unless something was done to reduce their open-loop gain at high frequencies. Most op amps are 'compensated' to be stable with unity closed-loop gain, but this compromises their performance in higher-gain circuits, so some are 'decompensated' and can only be used in circuits above a certain gain level (eg, 3-5×).

Some op amps are uncompensated, and the user must add component(s) to provide compensation (usually including at least one capacitor).

Our detailed discussion of op amp feedback and circuit stability in recent months, particularly with respect to problems caused by capacitive loads, leads to the fact that we often need to improve op amp stability by modifying a circuit. This usually involves adding zeros or poles to the loop gain.

Last month, we looked in detail at compensation for capacitive loading-induced instability using an isolation resistor – a resistor in series with the op amp's output, outside the feedback loop (see Fig.1).

This approach adds a zero to the loop gain, which improves stability by reducing phase shift in comparison with the uncompensated case. It is straightforward to implement, but can cause gain inaccuracy as it's outside the feedback loop.

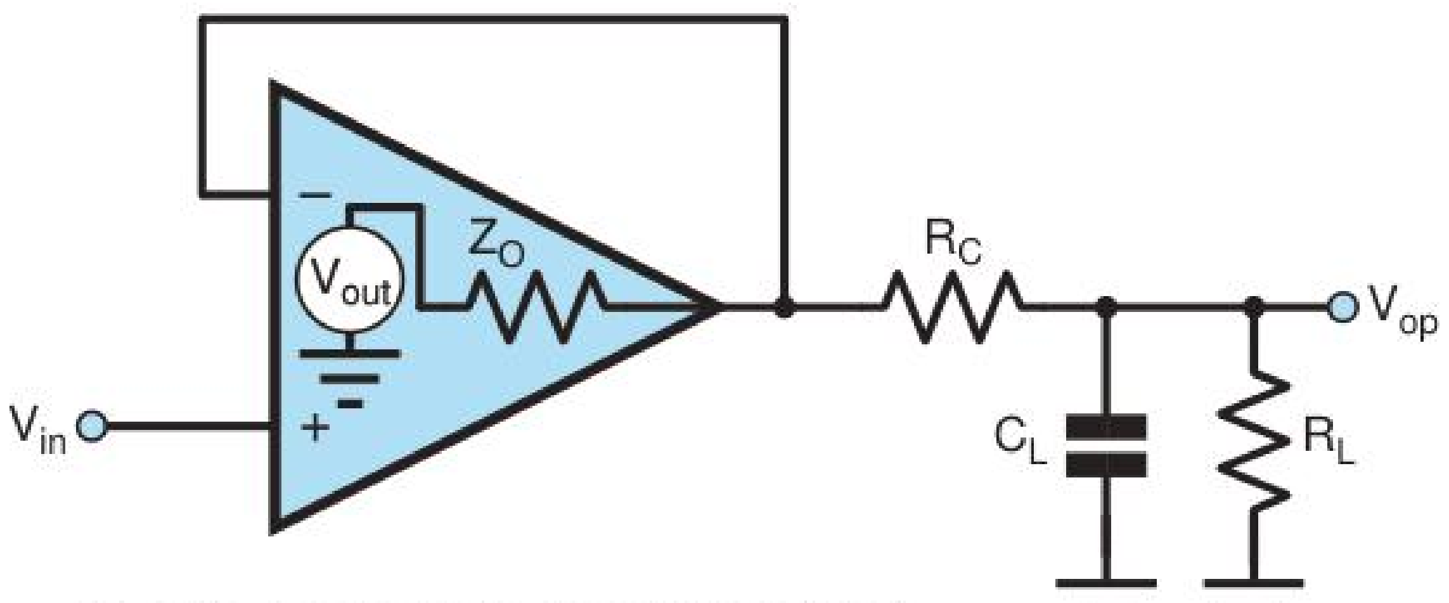


Fig.1: isolation resistor ($R_C$) compensation for improving stability.

The op amp's output impedance ($Z_O$ in Fig.1) is often ignored because the effective output impedance of a closed-loop op amp amplifier is typically very low due to the action of negative feedback. However, it plays a role in circuit stability – it appears in the loop-gain equations for circuits such as that in Fig.1.

To simplify circuit analysis, we may assume that an op amp's output impedance is purely resistive and constant, but in real devices, it varies with frequency and may behave capacitively or inductively over some frequency ranges. Last month, we discussed output impedance in detail and described how to obtain plots of op amp output impedance against frequency using LTspice.

### In-loop compensation

Using an isolation resistor for compensation results in the gain of the whole circuit no longer matching the designed value (no longer unity for a buffer, or the values set by the feedback resistors for other amplifiers).

This is because the isolation resistor forms a potential divider with any load resistance ($R_L$ in Fig.1), which reduces the amplitude of the output and hence the effective gain of the circuit. If precision gain is important, this is a major disadvantage.

Another way to think about this is that the circuit's output is not fully under the influence of the feedback, so it is not able to respond to load changes, which may lead to degraded circuit performance.

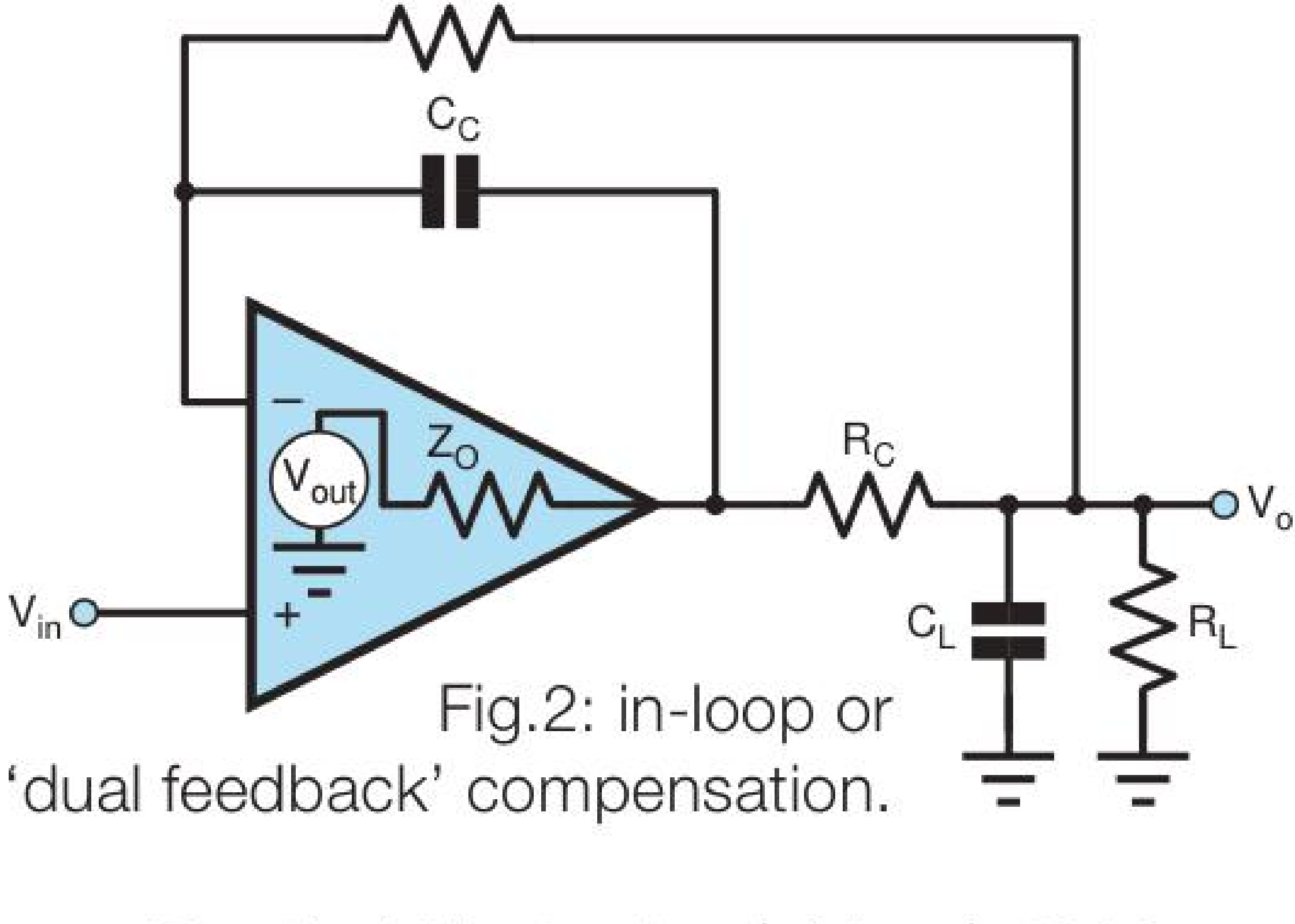


Fig.2: in-loop or 'dual feedback' compensation.

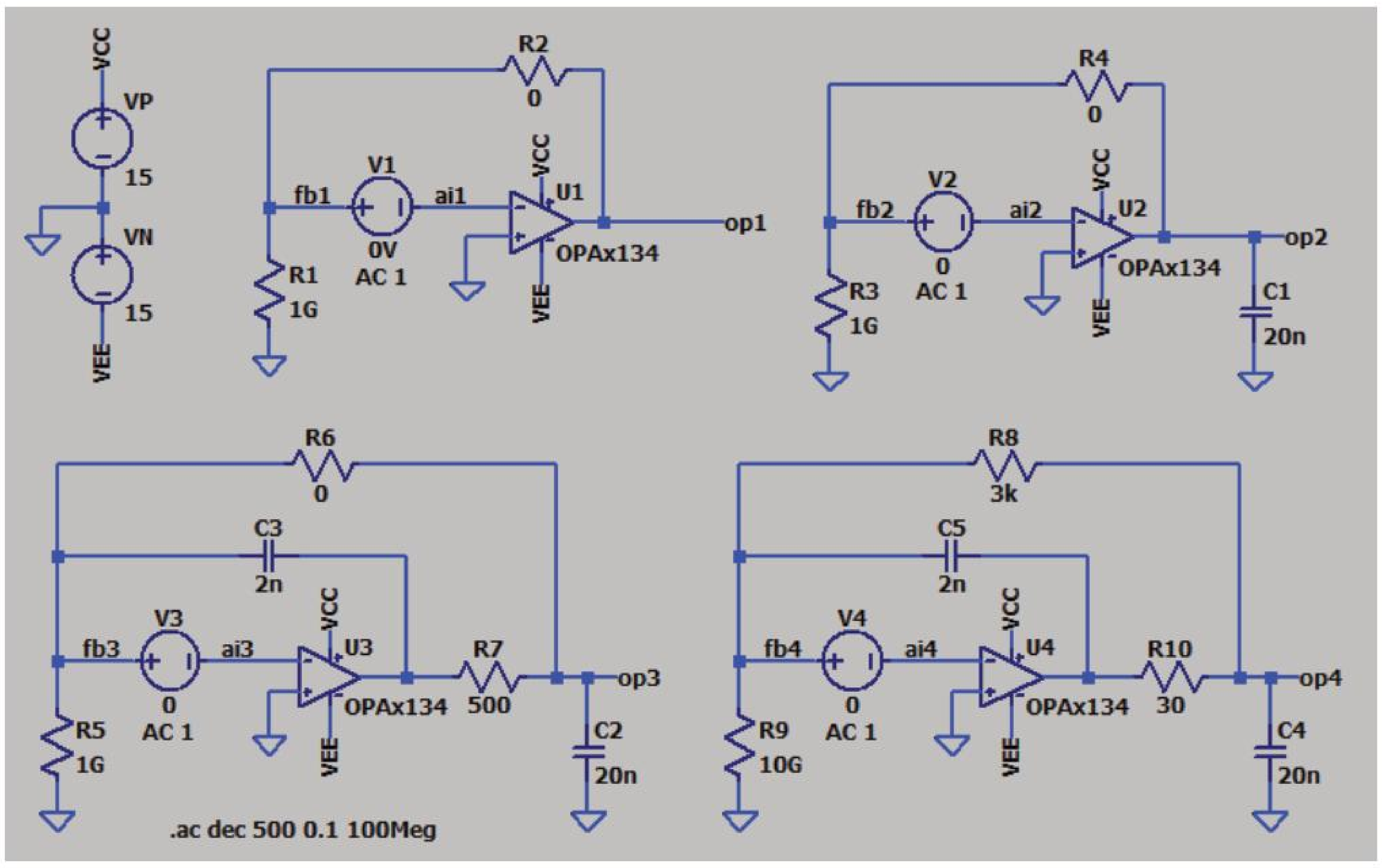


Fig.3: an LTspice schematic for investigating the efficacy of in-loop compensation.

A potential solution to this problem is to include the isolation resistor used for compensation ($R_C$) inside the feedback loop (see Fig.2), which is called in-loop compensation. As a feedback capacitor for compensation ($C_C$) is also required, there are two feedback paths, so the technique is also referred to as 'dual-feedback compensation'.

The circuit in Fig.2 is a unity-gain buffer, like the circuit in Fig.1, but we have included a feedback resistor ($R_F$). The resistor does not change the basic closed-loop gain. Ignoring the feedback capacitor (ie, the situation at low frequencies), this circuit still has 100% feedback via the resistor as there is no grounded resistor, which would be needed to set higher closed-loop gains.

The use of a feedback resistor can provide better compensation than just using the typical direct connection from the output to the inverting input, including when isolation resistor compensation is used, as in Fig.1.

Considering the simpler case, the key effects of this approach, with $R_F = 0\Omega$ and suitable values for the other components, are to add a relatively low-frequency pole, plus a zero above the pole, to the transfer function. These have the potential to improve stability. The circuit also produces a higher-frequency pole in the loop gain, which is generally of less importance.

The greater complexity of the in-loop compensation means that the loop-gain equation is quite large and cumbersome, so we will not go into the full algebraic analysis and derivation of the equations here. Ignoring the op amp's output impedance ($Z_O$) and load resistance ($R_L$), we can obtain a simplified, approximate expression for the pole frequency as:

$$f_p \approx \frac{1}{2\pi(R_C)(C_L+C_C)}$$

We can use the parallel combination of $R_C$ and $R_L$ in place of $R_C$ in the above if we need to take account of the load resistance. The pole can act similarly to the dominant pole used in the built-in compensation in most op amps; that is, lowering the gain as frequency increases to get to unity loop gain at a lower frequency. The zero frequency is:

$$f_z = \frac{1}{2\pi R_C C_C}$$

The zero has a similar role as in the basic isolation resistor approach – it reduces phase shift at frequencies above the pole. We could use $R_C + R_O$ instead (where $R_O$ represents a purely resistive version of $Z_O$) or just $R_C$ if necessary. If we include $R_F$, we get another zero in the response, which provides more compensation for the phase shift.

## Choosing component values

The choice of values for the components for in-loop compensation ($R_C$ & $C_C$) is not completely straightforward due the complex interactions in the circuit – the feedback has poles and a zero, while the op amp has poles and an output impedance that varies with frequency. Therefore, it may be necessary to experiment to find optimal values.

Simulations (like those used to illustrate this article) can be used to rapidly assess different value combinations before trying a physical implementation.

The isolation resistor ($R_C$) should be as small as possible; values in the 10s to 100s of ohms are often used. Ideally, $R_C$ should be smaller than the load resistance ($R_L$), maybe by a factor of 10 or more.

Op amp manufacturers may cover capacitive loading in device datasheets or other technical documents where values are suggested for specific devices. Where a feedback resistor ($R_F$) is used, it should be much larger than $R_C$ (maybe 100 times or more) to reduce interaction with the isolation resistor.

The frequency of the zero ($f_z$) is typically set to be about a decade below the unity gain (0dB) frequency of the uncompensated loop gain. Given that the loop gain is likely to be dropping at 20dB per decade at this point, we can also say that the zero should be close to the frequency at which the loop gain is 20dB.

With values of $R_C$ in the range mentioned above, this gives a starting point for $C_C$ using the $f_z$ equation above. The pole frequency is typically about a decade below the zero frequency. Comparing the $f_p$ and $f_z$ equations above, this implies that the load capacitance is about ten times $C_C$, which provides another way to choose a possible value for $C_C$.

## Component values with $R_F$ > 0Ω

The preceding discussion provides useful advice for selecting values but these are not strict rules. However, an application note from Intersil (now Renesas; see https://pemag.au/link/acb8) provides design formulae for circuits where $R_F$ is included. The application note is for their ISL7xx44 family of wideband amplifiers but should be more widely applicable. Their formulae for $R_C$ and $C_C$ are:

$$R_C \geq \frac{1+\sqrt{1+0.8\pi C_L B R_O}}{0.4\pi C_L B}$$

$$C_C = \frac{10 R_C C_L}{R_F}$$

Here, $B$ is the gain-bandwidth product of the amplifier and $R_F$ is chosen to be 100 or more times $R_C$. The first of these formulae is based on the decade below unity-gain rule mentioned above, while the second formula is based on setting the zero associated with $R_F$ a decade below that.

## Simulation examples

The LTspice circuit shown in Fig.3 can be used to investigate the loop-gain responses of an op amp unity-gain buffer with in-loop compensation. This example uses the OPAx134 op amp, which should be familiar from previous examples. Likewise, the simulation setup is as used in previous loop-gain examples and was discussed in detail in the February 2026 issue.

In Fig.3, U1 is an unloaded circuit, while U2 is loaded with a 20nF capacitive load – this circuit was previously shown to be unstable. The U3 circuit has in-loop compensation but without a feedback resistor (the feedback

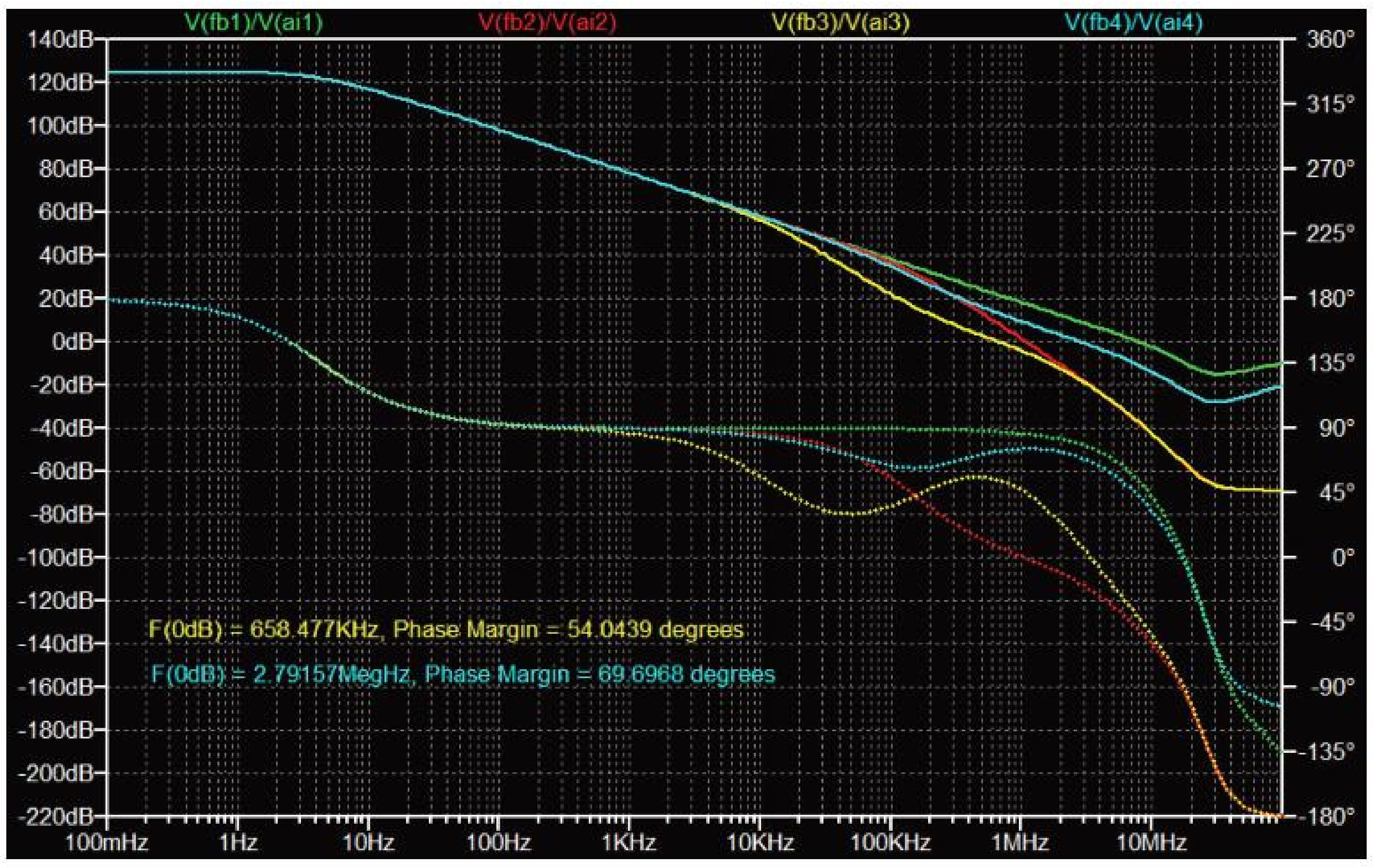


Fig.4: the results from simulating the circuits in Fig.3.

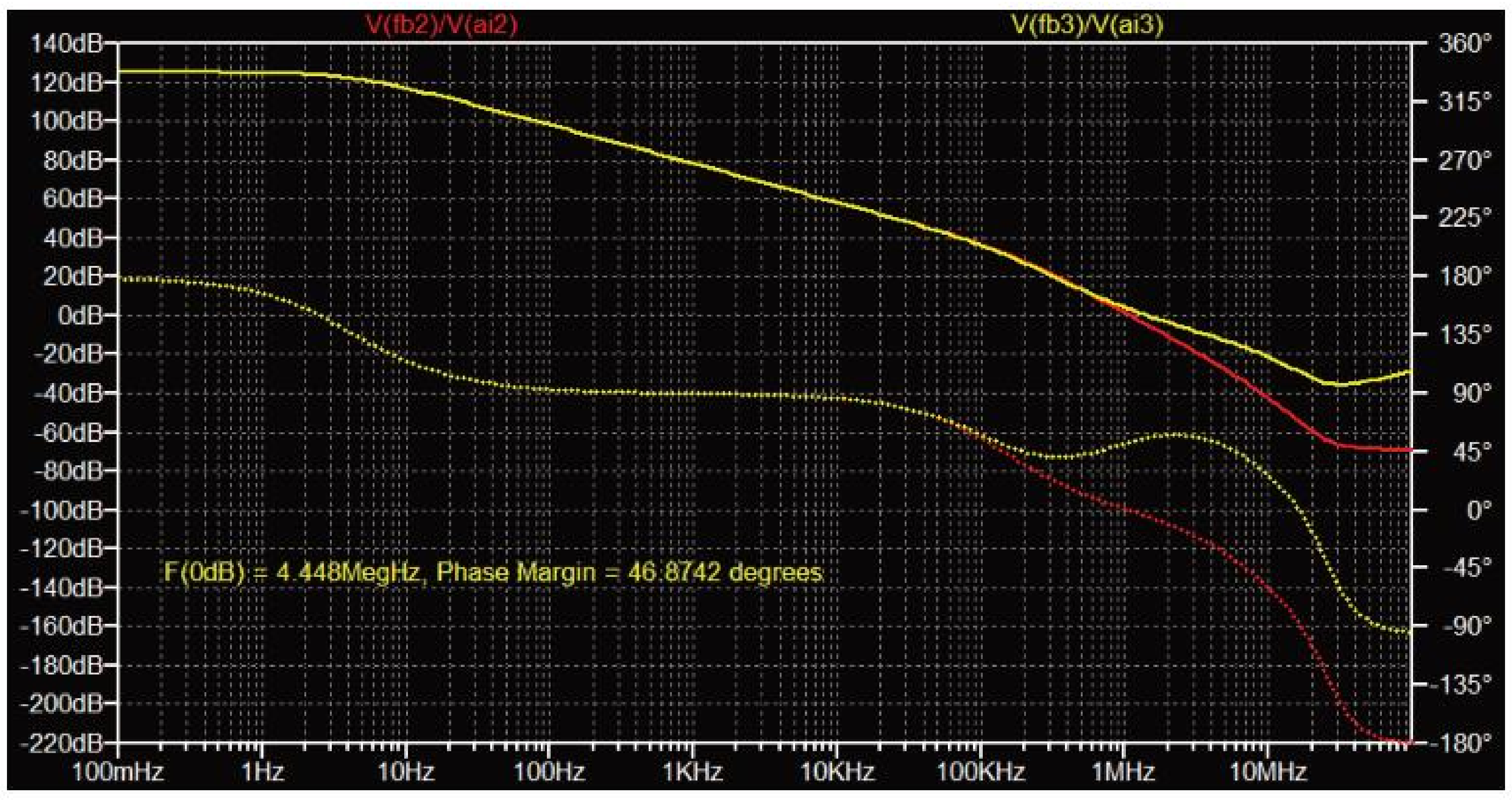


Fig.5: results from using out-of-loop isolation resistor compensation for op amp buffers from last month.

is a direct connection to the inverting input, as for U1 and U2).

From our simulation of the same circuit last month, we know the 0dB frequency of the loop gain is around 1.1MHz, so a zero in the low hundreds of kilohertz should be about right.

For the $R_C$ and $C_C$ values shown in Fig.3, the $f_z$ formula above gives $1 \div (2\pi \times 500\Omega \times 2{\times}10^{-9}\text{F}) = 160\text{kHz}$, which fits the guideline discussed above. The value of $C_C$ is about a tenth of $C_L$, as expected. The zero frequency is about $1 \div (2\pi \times 500\Omega \times [20{\times}10^{-9}\text{F} + 2{\times}10^{-9}\text{F}]) = 15\text{kHz}$ using the $f_P$ formula above.

The U4 circuit includes a feedback resistor. Using the Intersil formulae with $C_L = 20\text{nF}$, $R_O = 54\Omega$ as discussed last month, and $B = 8\text{MHz}$ from the OPAx134 datasheet, we get $R_C \approx 30\Omega$. Setting $R_F = 100 \times 30\Omega = 3\text{k}\Omega$, we get $C_C = 2\text{nF}$ (the same as the value used in the U3 circuit). It is worth noting that the isolation resistor is significantly lower in value in this circuit.

Fig.4 shows the simulation results with plots of the loop-gain frequency response. The green trace is the unloaded response from the U1 circuit while the red trace is the unstable response from the loaded but uncompensated U2 circuit. The yellow trace in Fig.4 shows the response of the compensated buffer without the feedback resistor (U3 circuit).

The cyan plot in Fig.4 shows the response of the U4 circuit with the feedback resistor. The response is much closer to the unloaded case than the U3 circuit without the resistor, and the phase margin is better, indicating that it is doing a better job of compensating for the load.

## Compared to last month's solution

For comparison, Fig.5 shows the results from the out-of-loop isolation resistor compensation of the same buffer from last month. The (uncompensated circuit) red trace in Fig.5 is the same as in Fig.4.

The yellow traces for the compensated circuits on the two graphs have a similar general shape, but have significant differences relative to the uncompensated response, which is due to the different compensation implementations.

The circuit with in-loop compensation has a pole at a relatively low frequency. In Fig.4, we see the phase (dotted yellow trace) decreasing with respect to the red (uncompensated) trace, starting at just over 1kHz. This is the effect of the pole at 15kHz (the phase shift effect starts about a decade below the pole frequency).

This early decrease in phase is not what we want for stability, but it is safe in this case because the phase shift remains sufficiently far above zero to not be a problem. As the frequency increases further, the zero at 160kHz reverses most of the shift due to the pole, keeping the phase shift well above zero.

In the out-of-loop case (Fig.5), the yellow and red phase traces track, including at about 20kHz, where the pole due to the load capacitance (which is at 146kHz) begins to affect the response. The curves separate at around 100kHz as the phase shift of the compensated circuit increases due to the zero caused by the isolation resistor.

The gain for the in-loop case (solid yellow in Fig.4) shows a reduction with respect to the uncompensated case in the range of 10kHz to 1MHz. A key point here is that the loop gain of the compensated circuit reaches unity (0dB) at a lower frequency than the uncompensated circuit. This is the advantage in stability provided by the low-frequency pole.

In comparison, the out-of-loop response (Fig.5) shows the gain decreasing more slowly for the compensated case. This is not ideal, but the phase increase due to the zero provides a stability improvement.

The differences and similarities between the in-loop and out-of-loop examples help to show that it is the specific interaction between the poles and zeros in a response that determines stability. Both examples show similar phase margins of around 50° but have different pole-to-zero relationships.

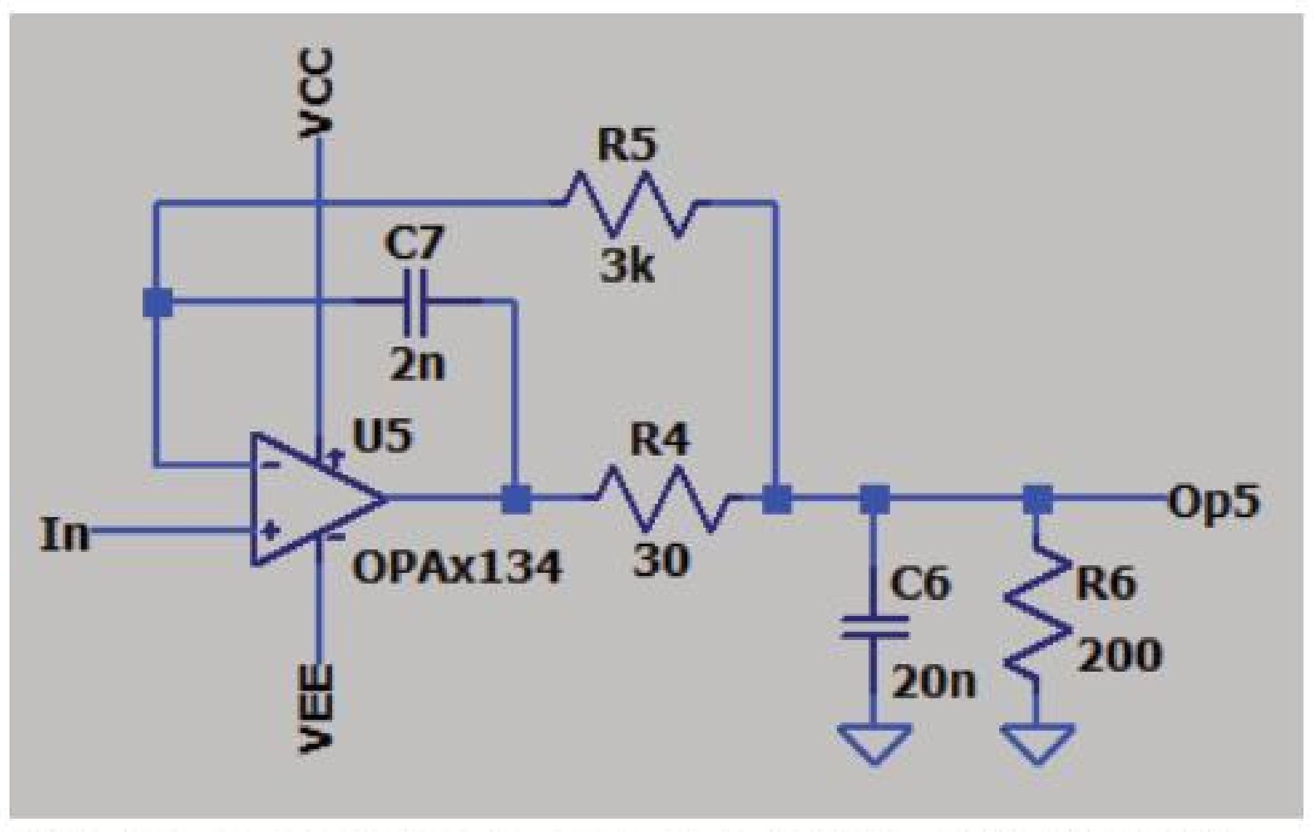


Fig.7: a compensated buffer circuit with a load resistance.

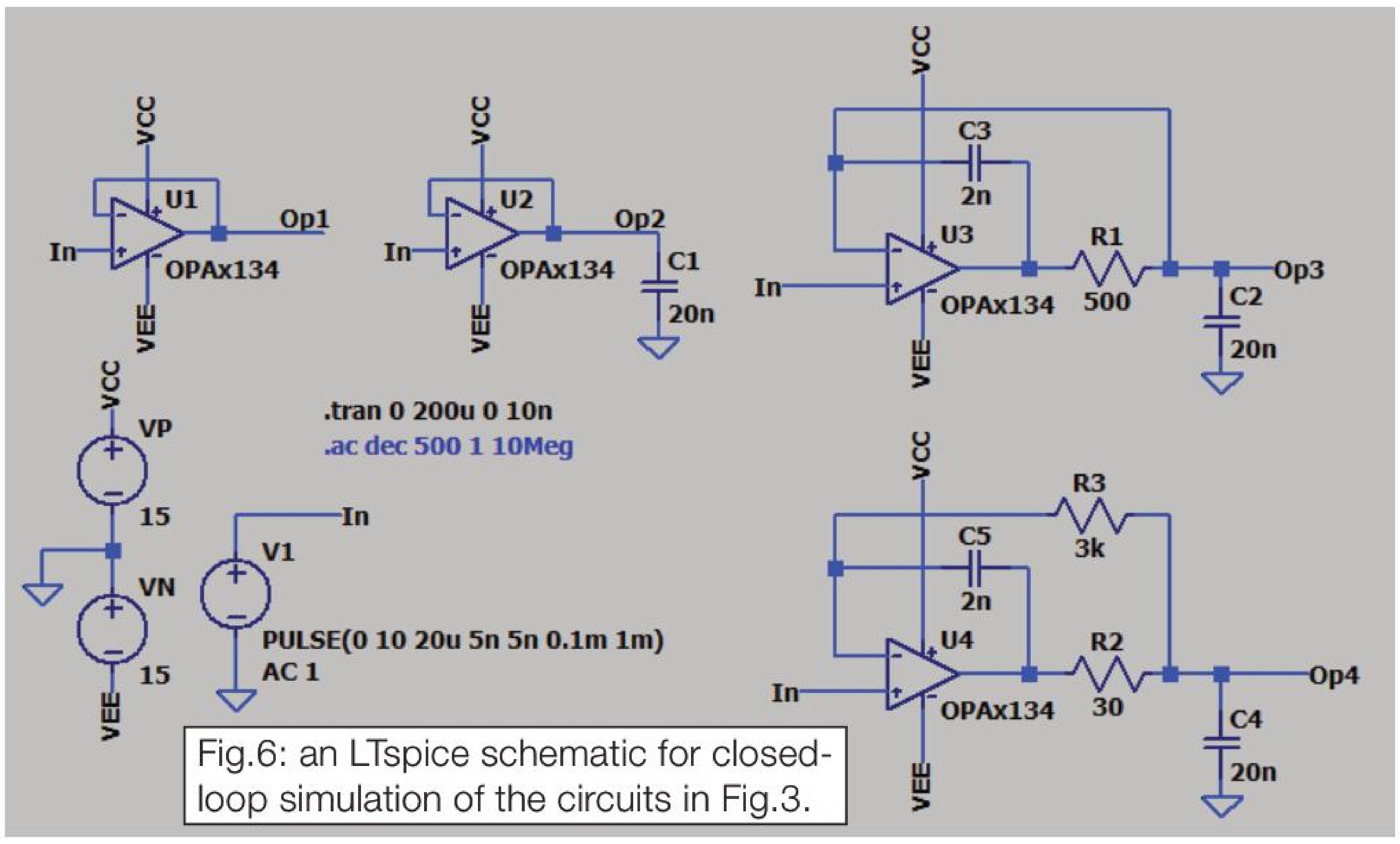


Fig.6: an LTspice schematic for closed-loop simulation of the circuits in Fig.3.

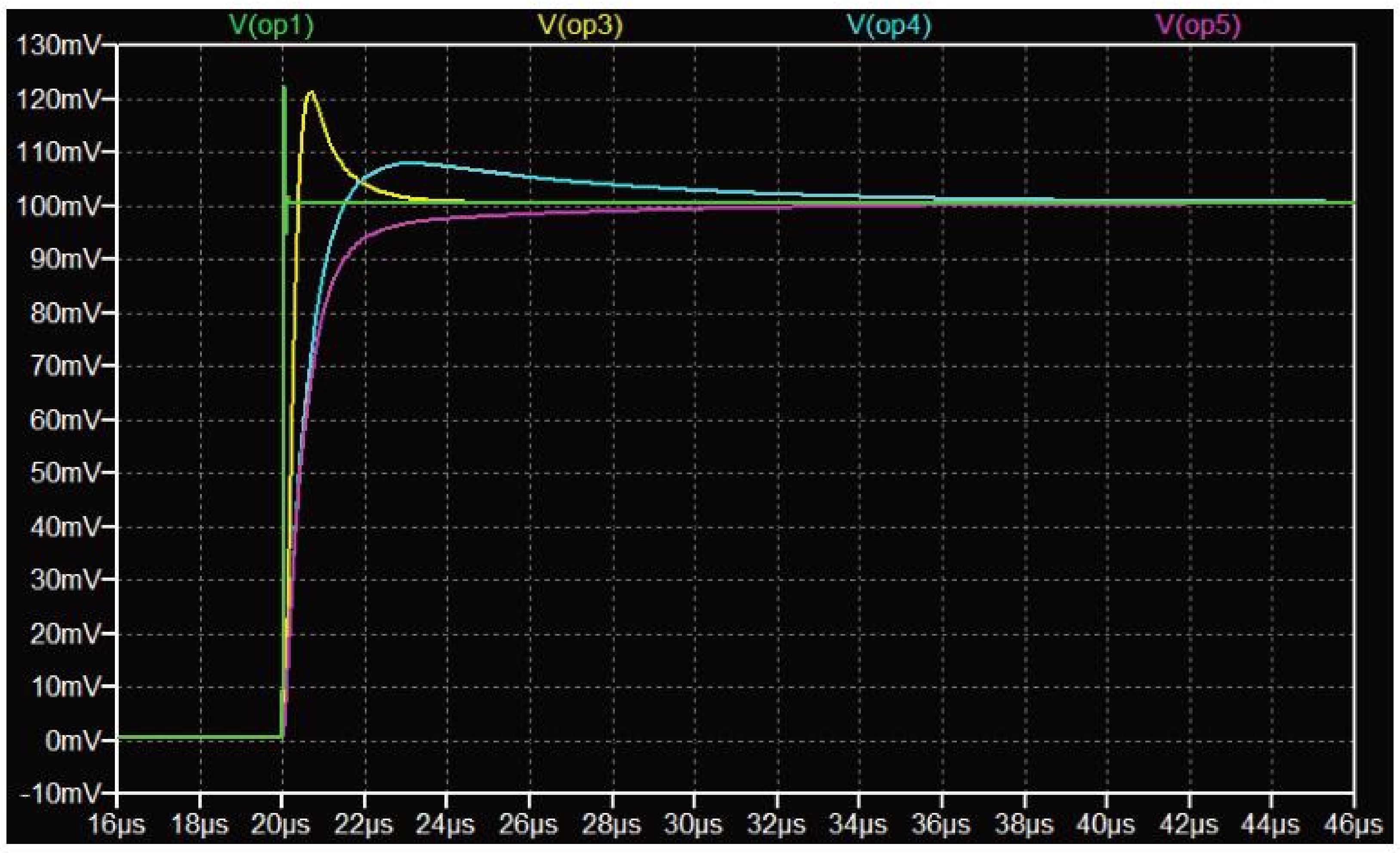


Fig.8: simulation results for the circuits in Fig.6 and Fig.7 with 100mV step input signals.

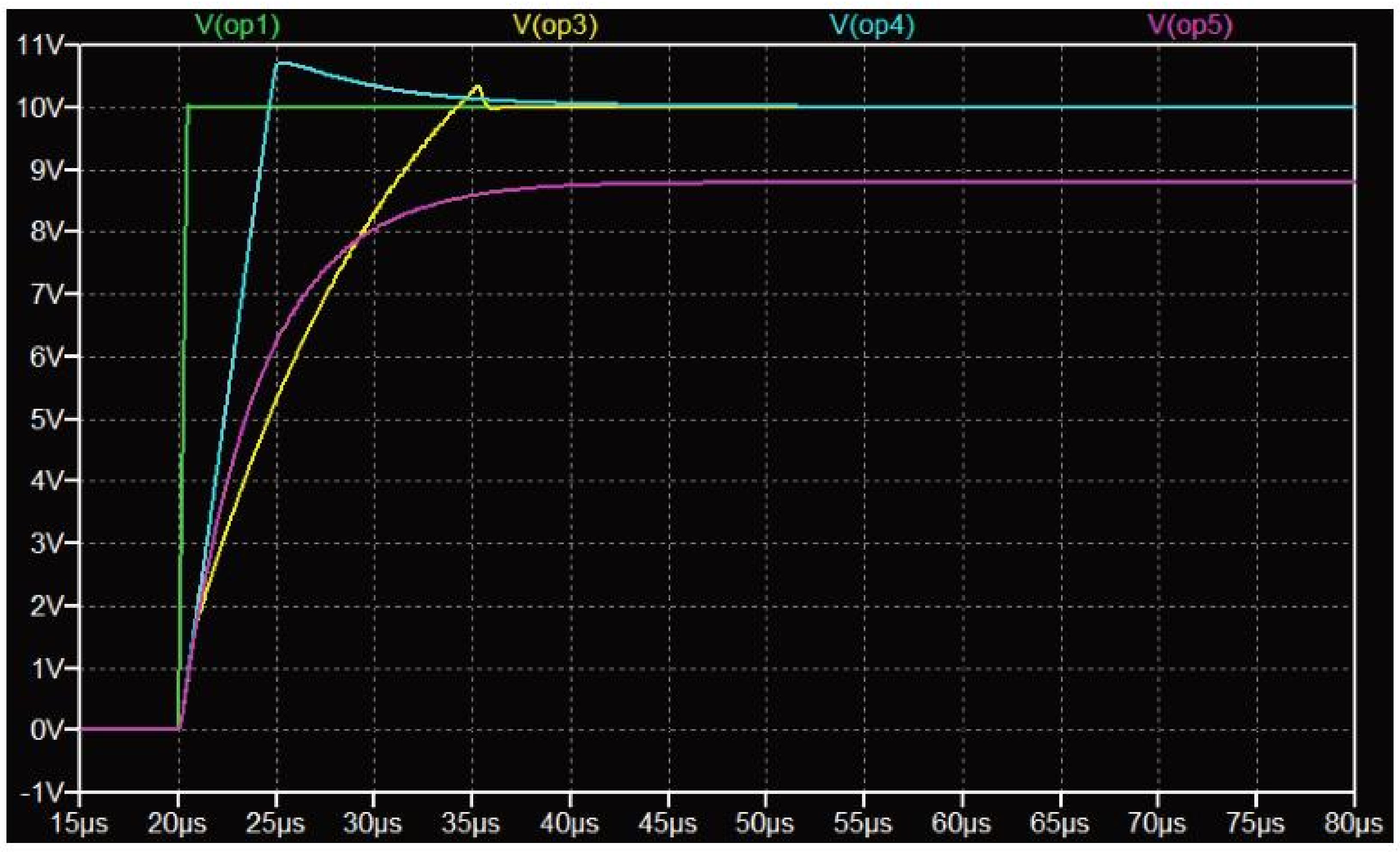


Fig.9: simulation results for the circuits in Fig.6 and Fig.7 with 10V step input signals.

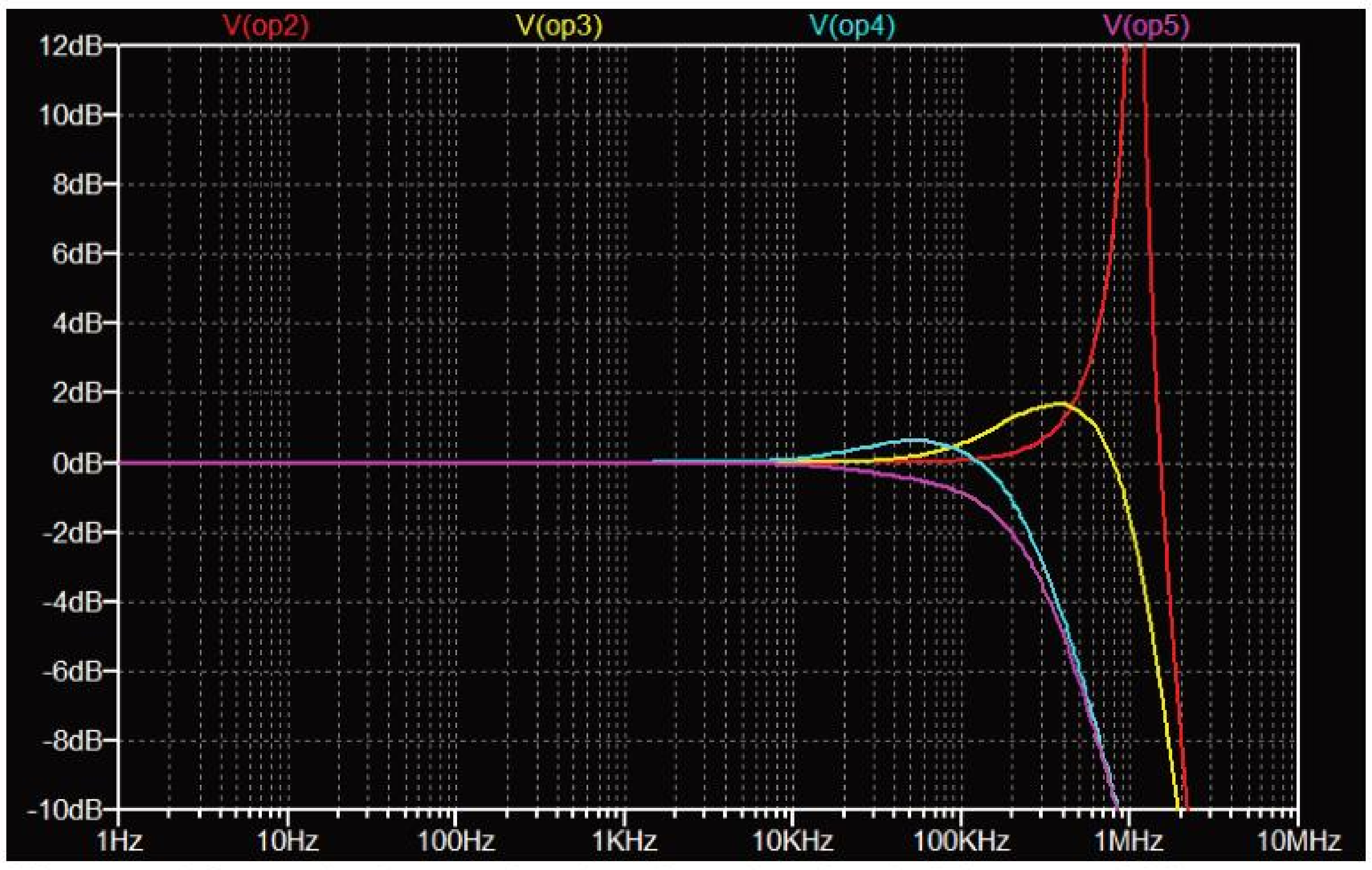


Fig.10: AC analysis results for the circuits in Fig.6 and Fig.7.

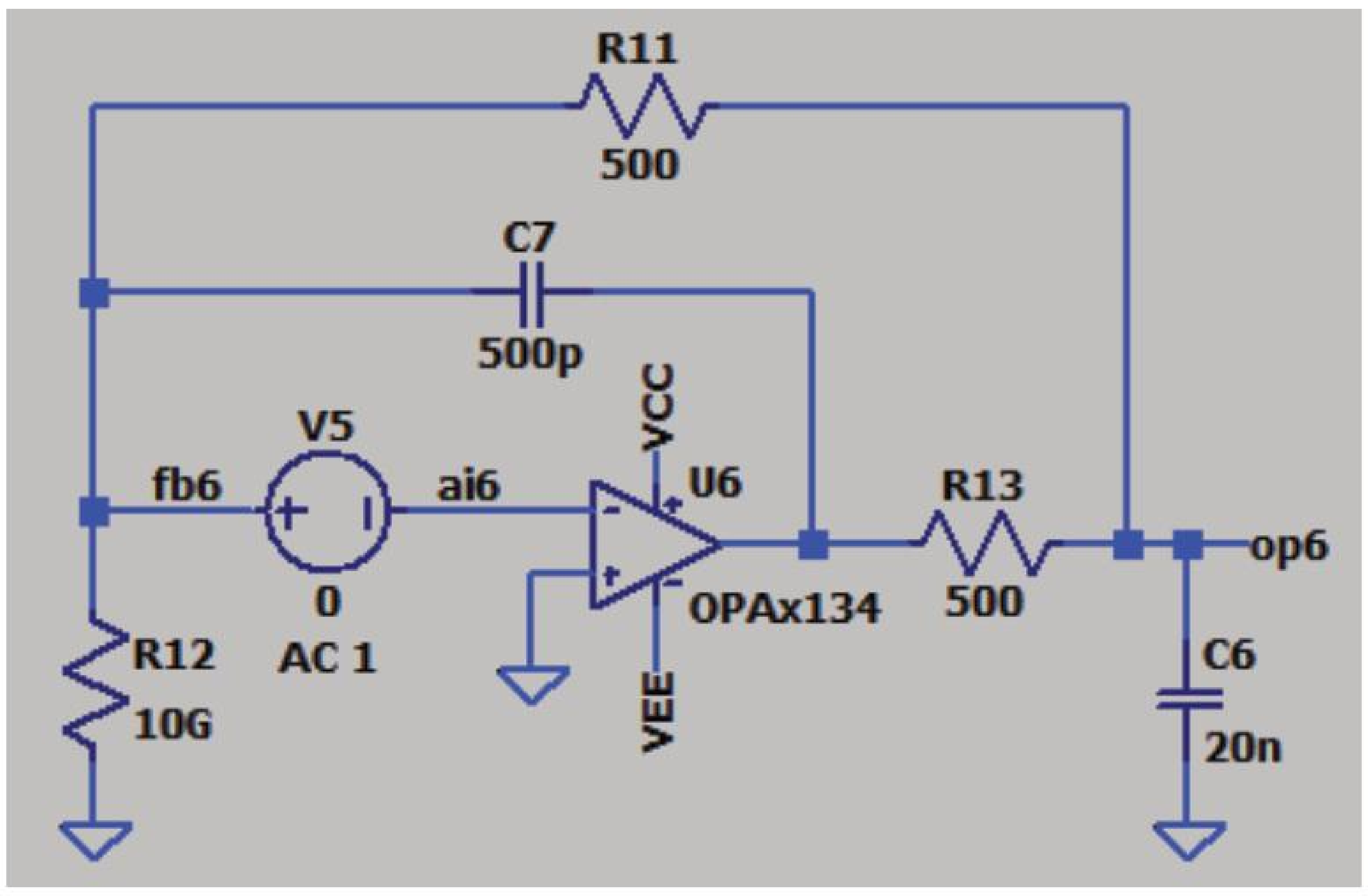


Fig.11: illustrating poor compensation component value choices.

## Closed loop simulations

Fig.6 shows an LTspice schematic for closed-loop simulation (normal circuit operation) of the circuits in Fig.3. The op amp instance names (U1 etc) and output names (op1 etc) are the same. Fig.7 shows an additional circuit where a resistive load has been added to the output.

The results for a transient simulation of these circuits with a 100mV step input are in Fig.8. As discussed in previous months, a step input will often trigger continuous oscillation in an unstable circuit and will result in large overshoots and ringing (decaying oscillation) in circuits with a poor phase margin. Therefore, they provide useful insights into stability.

The trace colours in Fig.8 are the same as those for Fig.4. The output of the unstable capacitively loaded circuit with U2 (red in Fig.4) is not shown, as the oscillations would swamp the graph. The results show a similar height overshoot for the unloaded circuit (green) and the one compensated without a feedback resistor (yellow). With the feedback resistor (cyan), there is a more damped response (little overshoot).

Unlike the out-of-loop compensation discussed last month, there is no gain error when a load resistance is present (see the circuit in Fig.7 and the magenta trace in Fig.8) – the output reaches 100mV, the same as the cases without resistive load.

Fig.9 shows the response of the same circuits with a 10V step input. In this case, the resistively loaded circuit produces an output level below the 10V expected from a unity-gain circuit. This looks similar to the gain error discussed last month but it is different. The problem is due to the op amp reaching its maximum output level.

The isolation resistor (R4 in Fig.7) and resistive load (R6) form a potential divider in the same way as they do for the out-of-loop feedback circuit, but because the feedback is taken from the final circuit output. The op amp adjusts the circuit output to the required level until it runs out of headroom.

The potential divider means that op amp output must go to a higher voltage to control the circuit output to the required level. So, for a given maximum op amp output, there will be a lower maximum circuit output.

The traces in Fig.9 show that the op amp can deliver 10V but cannot generate a sufficient voltage to obtain 10V from the potential divider formed by the isolation resistor and the resistive load in this example.

## AC analysis

Fig.10 shows the AC analysis results for the closed-loop gains of the circuits in Fig.6 and Fig.7. The traces have the same colour as in Fig.4. The gain response of the uncompensated capacitively loaded circuit (red trace) shows a large peak, typical of an unstable circuit.

The two compensated circuits without resistive loads (cyan and yellow traces) have much smaller peaks. The circuit with the resistive load (magenta trace) has the correct (unity) gain at low frequencies, like the other cases. This confirms the accurate gain, also visible in Fig.8. A similar graph last month for out-of-loop isolation resistor compensation showed the gain error inherent in that approach.

## Bad values

The circuit in Fig.11 is an addition to the schematic in Fig.3. It is a version

### Simulation files

Most *Circuit Surgery* columns feature the use of the free circuit simulation software LTSpice. It is used to support descriptions and analysis in *Circuit Surgery*.

The examples and files for this issue are available for download: https://pemag.au/link/acb7

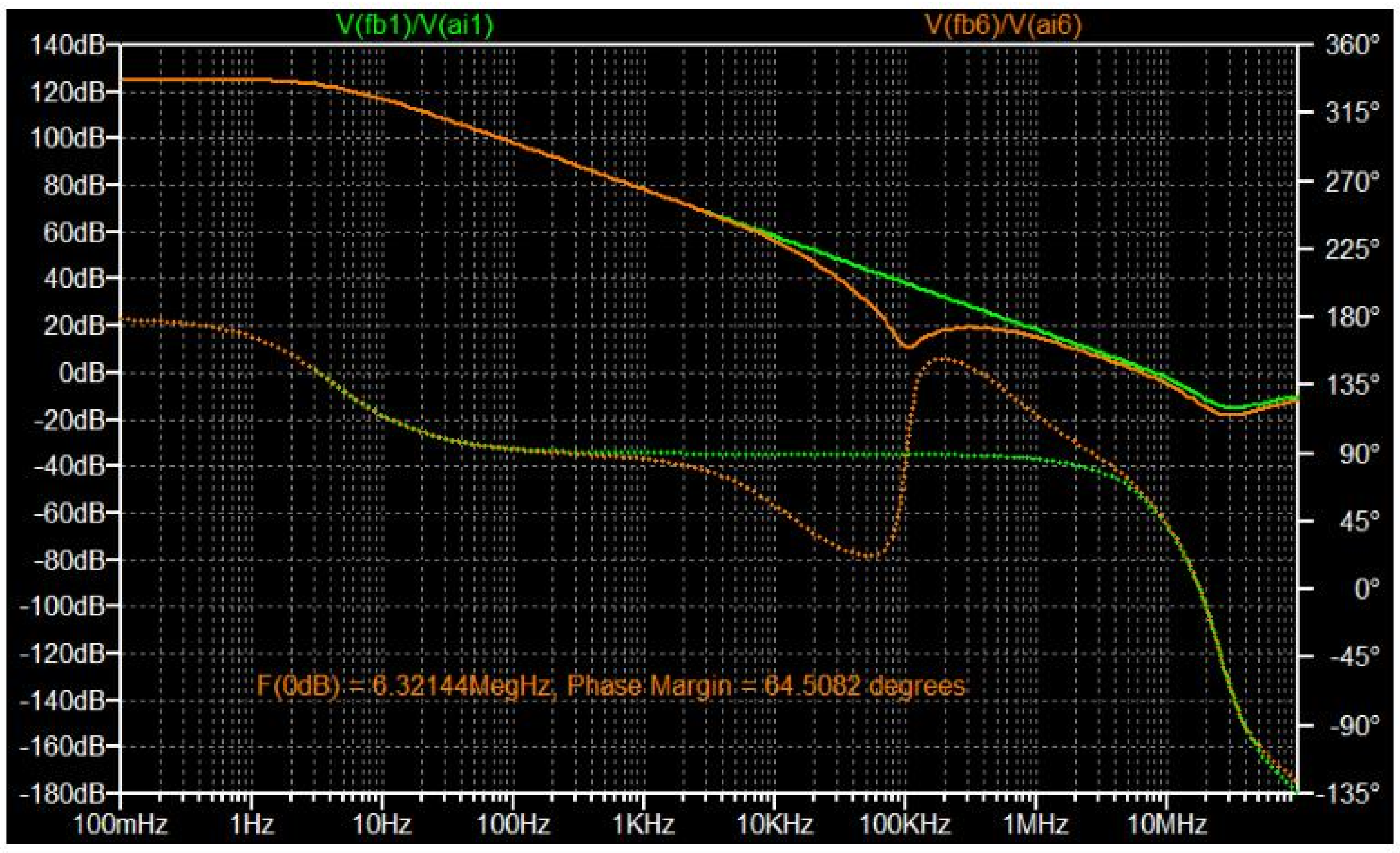


Fig.12: the simulation results for the circuits in Figs.3 & 10.

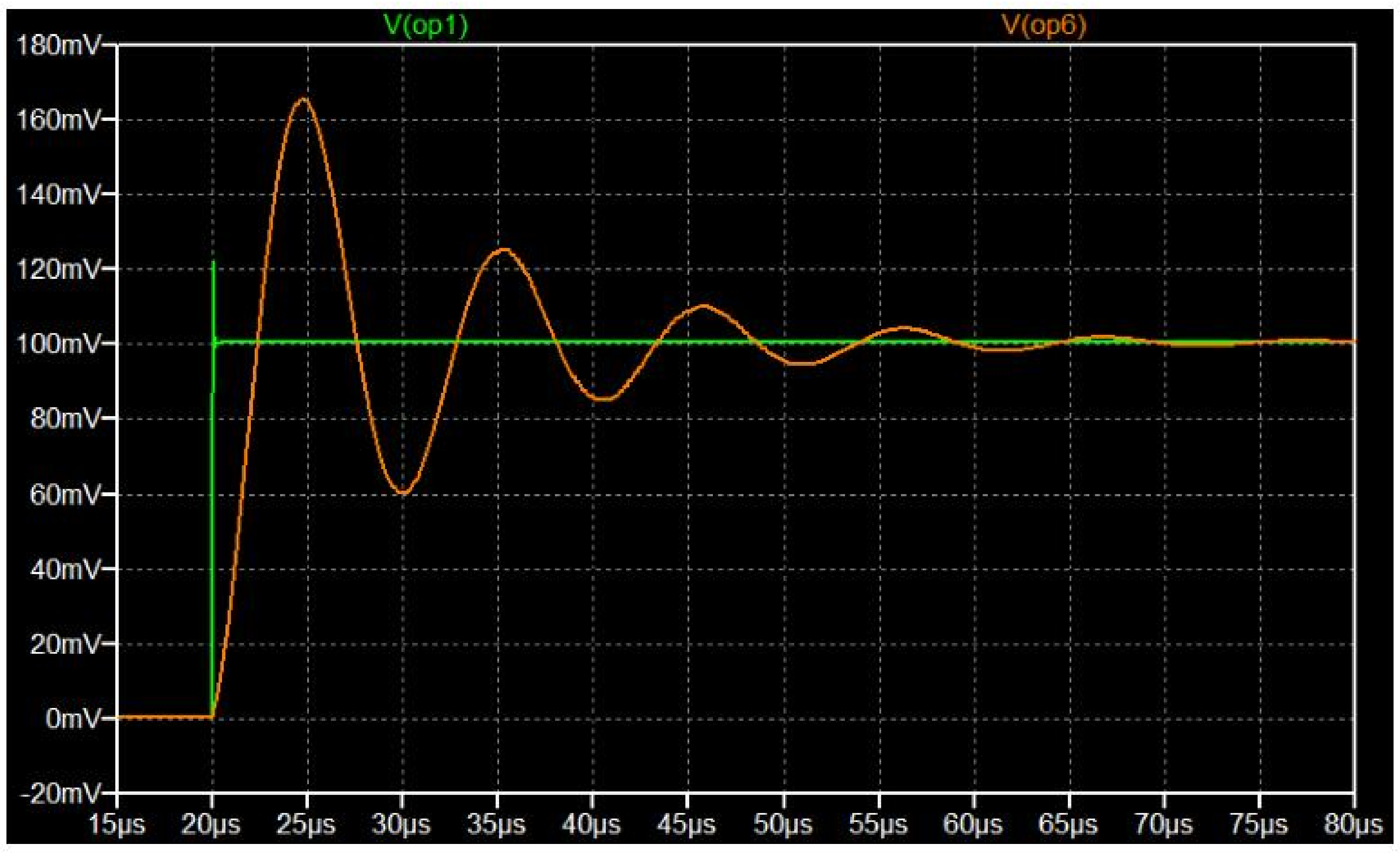


Fig.13: a transient step response simulation of the Fig.11 circuit.

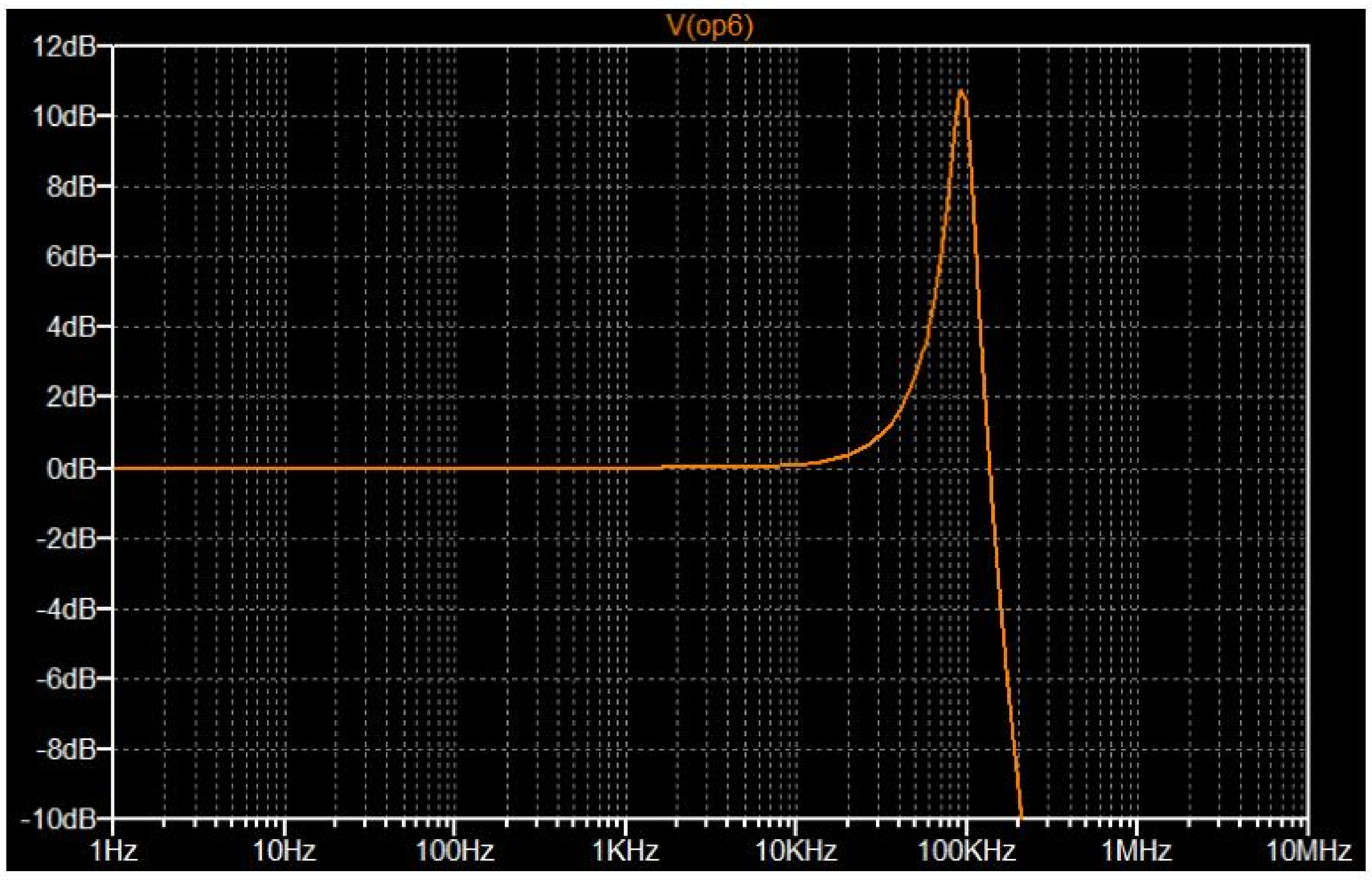


Fig.14: an AC analysis simulation of the circuit in Fig.11.

of the in-loop compensated buffer with a feedback resistor, but with different compensation component values. The loop-gain response is shown in Fig.12 (orange trace) along with the unloaded case (green, same as in Fig.4) for reference.

The response has some interesting features; its shape is somewhat different from those in the other responses we have looked at.

There is a notch in the gain curve around 100kHz – the gain starts to decrease more rapidly and then quickly changes direction and increases again before returning to match the unloaded gain curve. The phase plot shows a large and very rapid change, also around 100kHz.

LTspice reports a phase margin of 64.5° (Fig.12), which should mean that the circuit has very good stability, but the gain notch and rapid phase change in the loop-gain response should act as a warning that there may be problems with this circuit.

Fig.13 shows a transient simulation of the step response. The large overshoot and ringing indicate that the circuit does not have good stability and would likely be unacceptable in many designs.

The results in Fig.13 were obtained from a copy of the U4 circuit in Fig.6 (with the op amp labelled U6 and output op6) using the values of compensation components and load as shown in Fig.11.

Fig.14 shows the closed-loop frequency response of this circuit with the same axis scales as Fig.10. Again, we have an indicator of poor stability – the peak is much larger than the compensated circuit responses in Fig.10, although it is much lower than the peak for the fully unstable circuit.

The results from the circuit in Fig.11 show that an apparently good phase margin is not always sufficient to guarantee stability. The values of the components do not conform to the design guidelines and formulae discussed above.

This example was produced to illustrate the fact that this compensation technique can give poor results with some combinations of component values, which could be inadvertently chosen. It demonstrates that you should check the closed-loop frequency response and step response, as well as phase margin, when designing compensation circuits.

The shape of the loop-gain and phase responses in Fig.12 indicates the presence of a pair of complex conjugate zeros.

This term relates to the s-domain (Laplace) analysis of circuits, which we introduced in the January 2026 issue and have used to analyse of some of the circuits discussed since. We have kept to simple cases; complex conjugates go beyond what we have covered.

In summary, the in-loop compensation technique overcomes the gain error problem with an out-of-loop isolation resistor, but is more complex and can suffer from a more limited output swing. The output swing concern is more likely to be a problem when using the op amp close to its minimum supply voltage, where its full swing is needed for the output.

Users of this approach should also be aware of the possibility of poor stability even if the phase margin is apparently good.

## Snubber network compensation

Another method of compensating capacitive loads is to connect a snubber network, comprising a resistor ($R_S$) and capacitor ($C_S$) in series, at the output of the op amp (see Fig.15).

Two months ago, we analysed the effect of a capacitive load on the op amp. To keep things simple, we assumed that $Z_O$ was resistive (as $R_O$) and ignored $R_L$. We found that the loop gain with the load capacitance is the loop-gain without the load (that is, $-\beta A_O(s)$) multiplied by the potential divider factor of:

$$\frac{1/sC_L}{R_O+1/sC_L}$$

If we include $R_L$, we need to replace both occurrences of $1 \div sC_L$ in the equation (the reactance of the load capacitor) with the parallel combination of $C_L$ and $R_L$.

This means the algebra involves a few more terms, but it is not too difficult.

After a few steps, we can get the equation in the form of divided polynomials in the *s*-domain. This is needed to obtain poles and zeros, as has been described in previous examples. Specifically, we get:

$$\frac{1}{1+R_O/R_L+sR_OC_L}$$

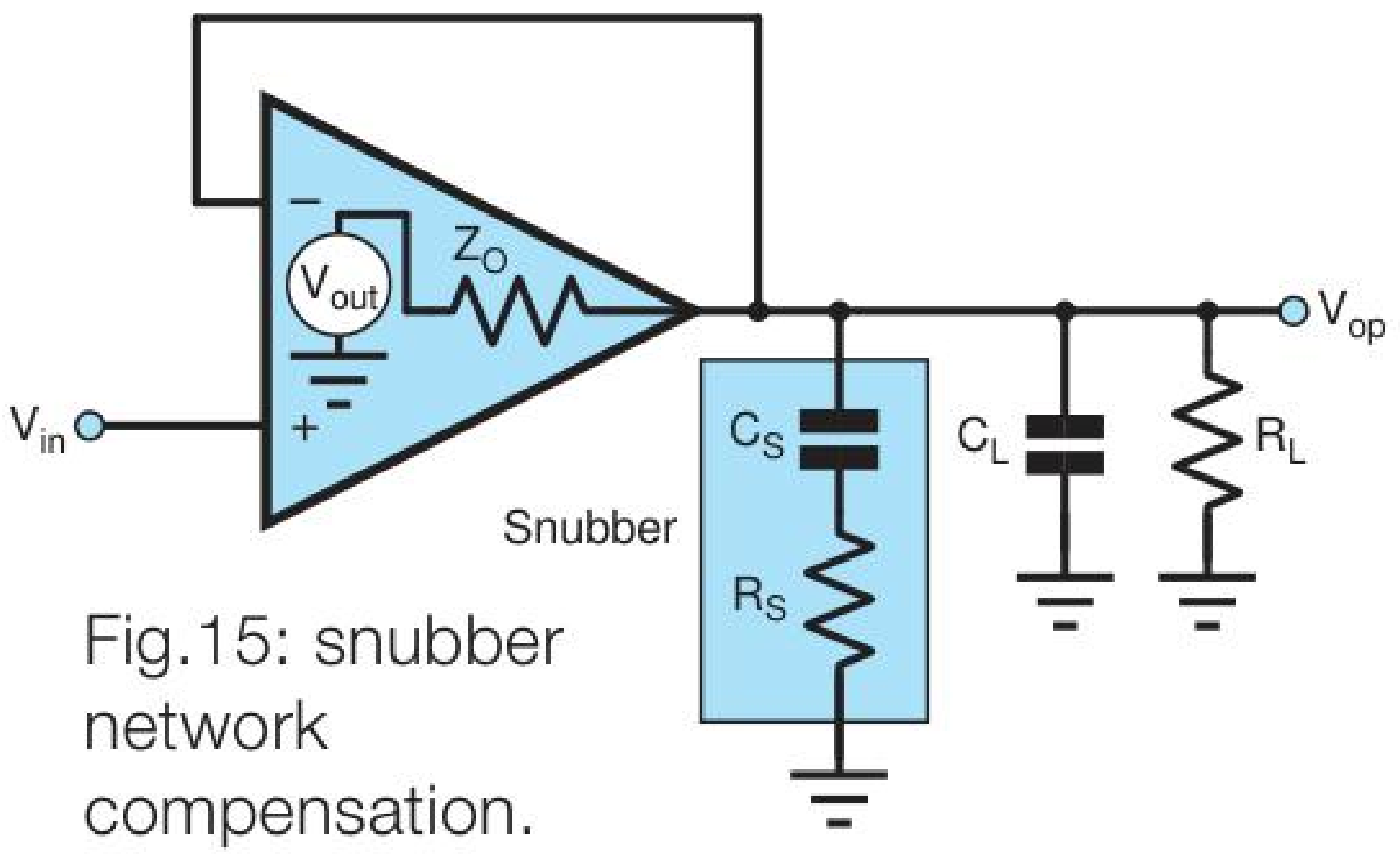


Fig.15: snubber network compensation.

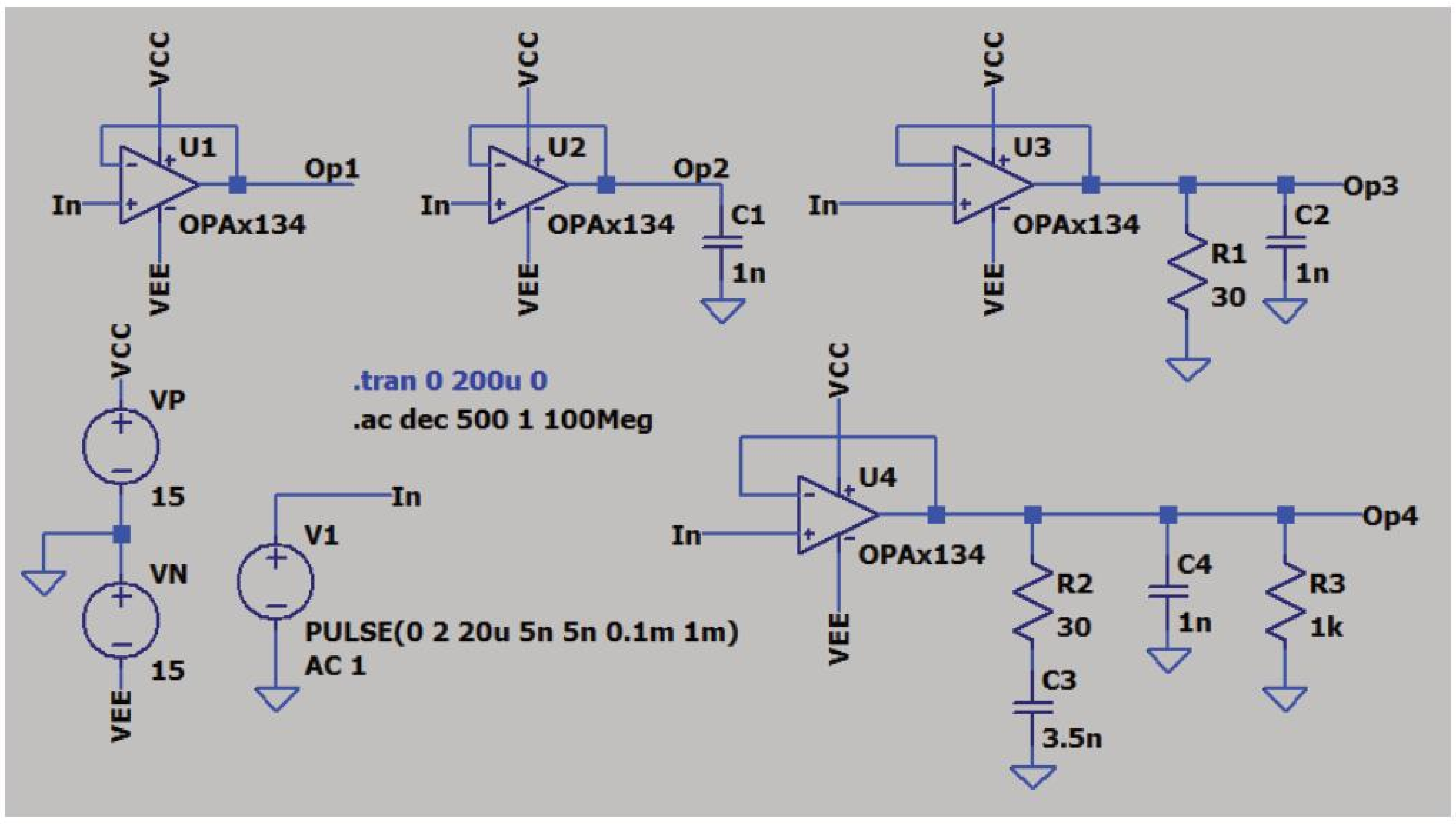


Fig.16: a set of LTspice circuits for investigating snubber network compensation.

As usual, to find the pole(s), we solve for $s$ with the denominator equal to zero. We then substitute $s = j\omega = j2\pi f$ and find the magnitude to get a pole frequency.

A trick we need here is to notice that it is possible to arrange the pole frequency equation so it contains the term $R_L R_O \div (R_L + R_O)$, which is the parallel combination of $R_L$ and $R_O$ (denoted as $R_L \,||\, R_O$). We can then write the pole frequency as:

$$f_p = \frac{1}{2\pi (R_O \| R_L) C_L}$$

It is this pole, due to the load capacitance, that increases the phase shift and causes instability.

If we add another, smaller resistor in parallel with $R_L$ (adding just the snubber resistor, $R_S$, and ignoring $C_S$ for the moment) then $R_S$ will be in parallel combination with $R_L$ in the pole frequency equation.

We don't have to work through all the algebra for the new pole frequency, because we can just replace $R_L$ with $R_L \,||\, R_S$ in the equation to get:

$$f_p = \frac{1}{2\pi (R_O \| R_L \| R_S) C_L}$$

This will be at a higher frequency, particularly if $R_S$ is small compared to $R_L$. A small $R_S$ will dominate the parallel resistance terms and increase $f_p$ due to the reciprocal relationship. Moving the loop-gain pole to a higher frequency will increase stability (in other words, provide a better phase margin) because the phase shift it causes will be at a frequency where the op amp's gain is lower.

If we only used a low-value resistor in parallel with the load, we would shift the pole as described, but we would also reduce the gain of the circuit at low frequencies. This would cause gain errors, or even make the circuit unusable.

## Limitations

Adding the snubber capacitor creates a zero at:

$$f_z = \frac{1}{2\pi R_S C_S}$$

With the capacitor in place, the snubber is effectively open-circuit at sufficiently low frequencies, so the snubber resistor has little or no effect there. At higher frequencies, the capacitor needs to have low impedance so that the snubber resistor loads the op amp, reducing the circuit's closed-loop gain.

Specifically, we need the loading effect to become active at around the frequency where peaking occurs, such as that seen in Fig.9 (red trace) and Fig.13. The resistor needs to load the op amp sufficiently to reduce the peaking enough to provide acceptable stability.

The larger $C_L$ and smaller $R_S$ are needed to overcome the effects of the load. Unfortunately, there are problems if $R_S$ is too small. A smaller $R_S$ requires a larger $C_S$, but if $C_S$ is too large, it will have too low an impedance at low frequencies, affecting the gain, which we don't want.

Also, if $R_S$ is too small, the op amp may go into protective current limiting driving it, which will limit the output voltage (to the current limit times $R_S$) and/or limit the rate of change of the output voltage, because limiting the current will limit the rate of change of the voltage on $C_S$.

The impact of these dependencies is to limit the maximum load capacitance that the snubber network can adequately compensate for a given op amp.

## Snubber component values

A trial-and-error approach to selecting snubber values was described in an Elantec application note from 1998 (**https://pemag.au/link/acb9**). Intersil/Renesas acquired Elantec Semiconductor in 2002. The procedure is as follows.

Find the peak frequency ($f_{pk}$) in the circuit's closed loop response, then load the amplifier with different snubber resistors, without the snubber the capacitor, until an acceptable reduction in the peak is obtained with resistor value $R_S$ (by simulation or measurement). $R_S$ is typically in the 10s to 100s of ohms.

The capacitor needs to have a lower impedance than $R_S$ (say, one third the value) at the peak frequency, so $R_S \div 3 = 1 \div (2\pi f_{pk} C_S)$. This is equivalent to making the snubber zero frequency three times lower than $f_{pk}$, so:

$$C_S = \frac{3}{2\pi f_{pk} R_S}$$

## Simulations

Fig.16 shows an LTspice schematic for investigating snubber network compensation. The setup is similar to previous closed-loop examples, with the U1 circuit being an unloaded buffer and the U2 circuit a capacitively loaded version with no compensation. The load capacitance (1nF) is smaller than in previous examples, in line with the limitations discussed above.

The U3 circuit includes a snubber resistor without a capacitor. This circuit is for investigating the effect of the resistor on the gain peaking as described above; it is not an implementation of the compensation. The U4 circuit has the snubber compensation (R2, C3) and a load resistor (R3) as well as the capacitive load (C4).

Fig.17 shows the peaking in the closed-loop response. The red trace (op2) is the loaded, uncompensated case with excessive peaking at 4.5MHz. The blue

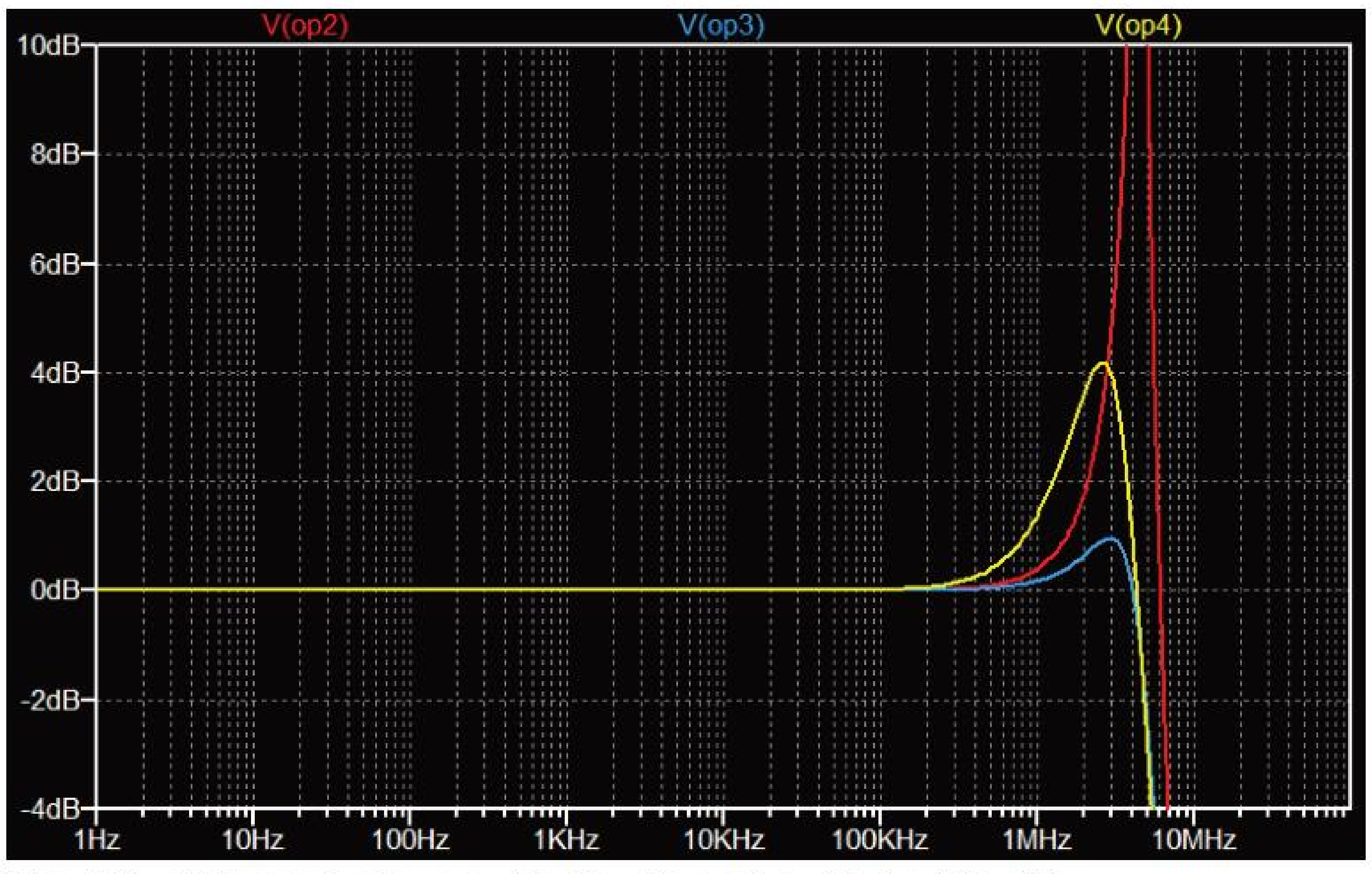


Fig.17: AC analysis results for the circuits in Fig.16.

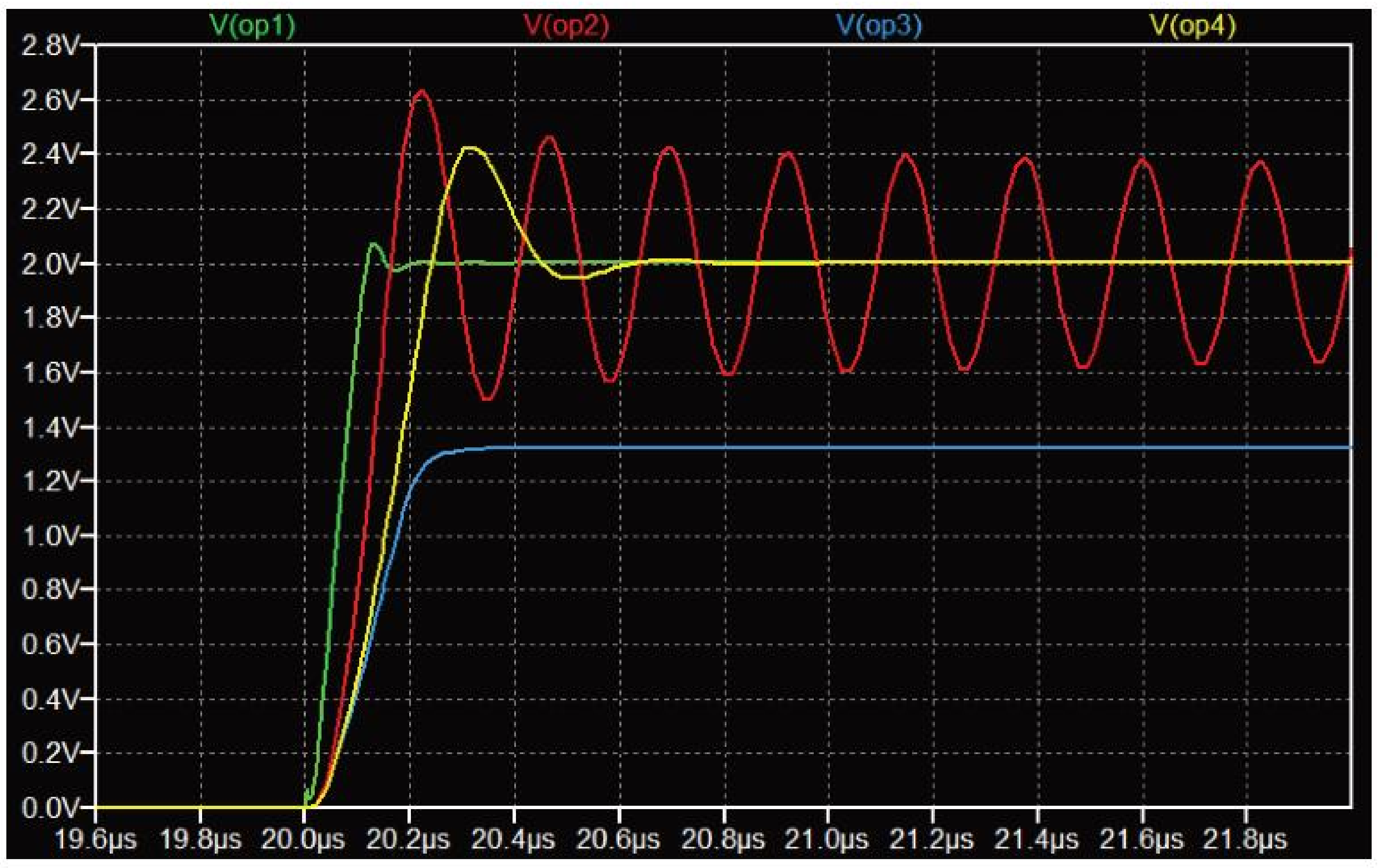


Fig.18: transient analysis (step response) for the circuits in Fig.16.

trace (op3) shows the effect of adding just a snubber resistor (R1 in Fig.16). The peak is reduced significantly.

Of course, the decision on what is acceptable depends on the context. We do not have a specific requirement in this discussion, so we are assuming that the response with the 30Ω resistor is deemed acceptable. Using the formula above, we get $C_S = 3 \div (2\pi \times 4.5{\times}10^6\text{Hz} \times 40\Omega) = 3.5\text{nF}$. The yellow trace (op4) is the response with the snubber capacitor in place.

Using the same approach as in Fig.3 and Fig.4 (and previous examples), we obtain a phase margin of 36° (not shown) for the U4 circuit with the snubber in Fig.16.

This is not above the commonly used rule-of-thumb minimum phase margin of 45°, but is significantly better than the 0.5° phase margin without the snubber, so it may be acceptable.

Fig.18 shows the step response for the unloaded (op1, green) and uncompensated (op2, red) cases. We can see significant oscillations with the capacitive load. These oscillations do eventually decay (visible using a longer timescale), but the circuit is very close to being fully unstable.

With the snubber in place (op4, yellow), there is still a fair amount of overshoot, but the oscillation decays quickly. The blue trace (op3) is for the U3 circuit with just the resistor. This shows the need for the capacitor to prevent gain reduction at lower frequencies.

## Feedback capacitor compensation

Another compensation technique that is commonly used is to place a capacitor ($C_F$) across the feedback resistor (see Fig.19). This could be seen as a simplified version of the in-loop approach, and as such may be less effective.

However, for amplifiers with above-unity gain (which already have a feedback resistor), it only requires one additional component and, as previously discussed, higher gain circuits are generally easier to stabilise.

As well as its potential use for instability due to capacitive loads, this technique is also commonly used to compensate for instability due to capacitance at the op amp's inputs. This input capacitance is typically a combination of the op amp's own input capacitance and the stray (parasitic) capacitance due to the wiring and components connected to the op amp's inverting input.

Parasitic input capacitance ($C_P$) creates a pole in the loop gain at:

$$f_p = \frac{1}{2\pi R_E C_p}$$

Here, $R_E$ is the equivalent parallel resistance of the gain-setting resistors (feedback and input or grounded resistor). Recall that for a load capacitor without load resistors, the relevant pole is at $1 \div 2\pi R_O C_L$.

Typically, $C_P$ is much smaller (a few to tens of picofarads) than the trouble-causing load capacitances ($C_L$) we have been looking at in the examples above, but the pole frequency depends on the gain resistors which are typically much larger than the op amp's output resistance ($R_O$), so the effects can be similar.

The feedback capacitor introduces a zero in the loop gain, which can counteract the instability-causing pole:

$$f_p = \frac{1}{2\pi R_F C_F}$$

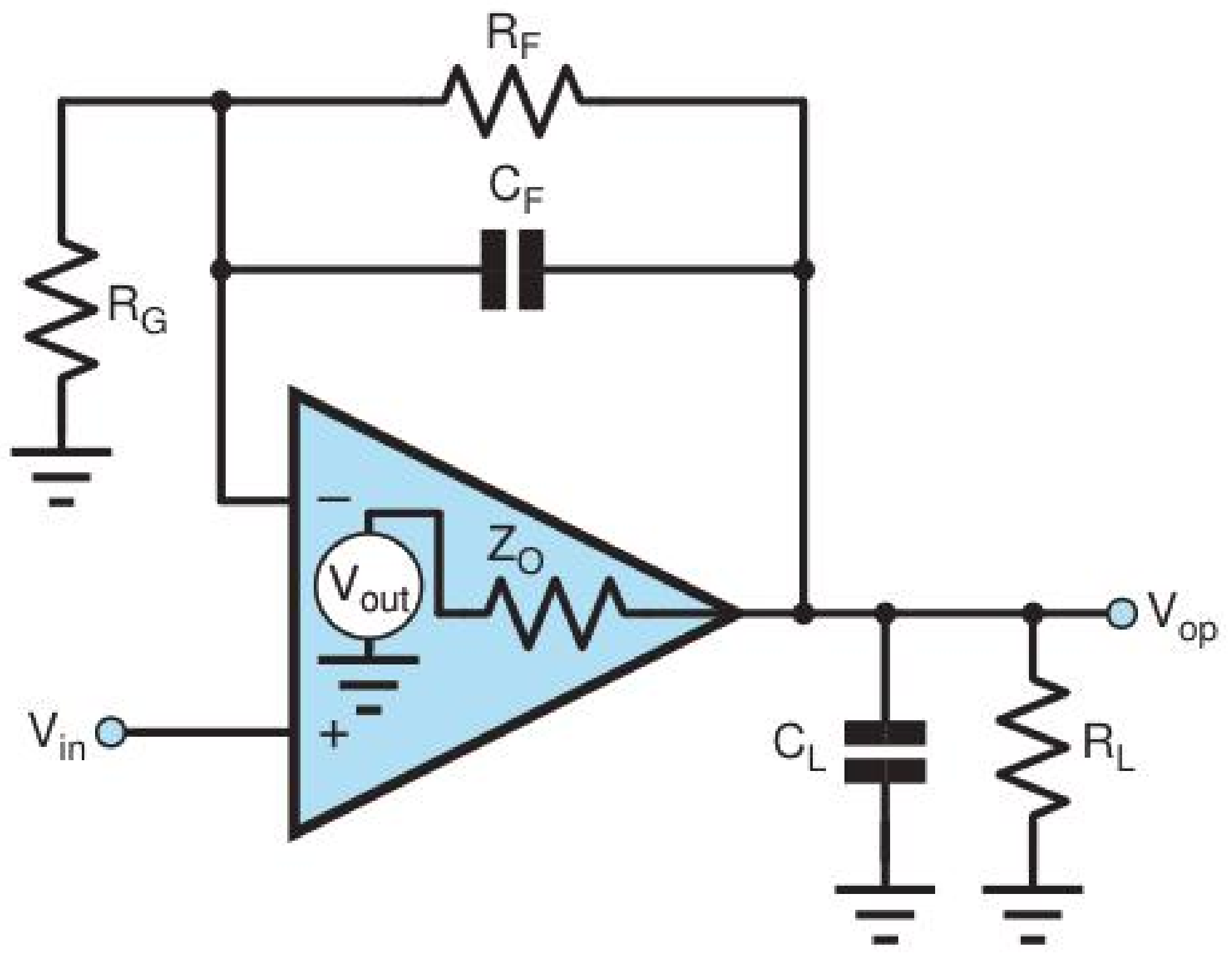


Fig.19: feedback capacitor compensation.

A starting point for component selection is to match the zero frequency from the compensation capacitor to the pole frequency from the load or parasitics. Some experimentation may improve the results, as the overall behaviour is complicated by the higher-frequency poles in the loop gain.

## Simulation example

Fig.20 shows an LTspice schematic for investigating feedback capacitor compensation with a similar setup to previous examples. The circuits are non-inverting amplifiers with a gain of 10. Assuming $R_O = 54\Omega$, as used previously, we get the pole due to the load of 30nF (C3) at $1 \div (2\pi \times 54\Omega \times 30{\times}10^{-9}\text{F}) = 98\text{kHz}$.

Setting the zero from the feedback capacitor at the same frequency requires $C_F = 1 \div (2\pi \times 98{\times}10^3\text{Hz} \times 9{\times}10^3\Omega) = 180\text{pF}$ (the value of C2 in Fig.20).

Using the same approach as in Fig.3 and Fig.4 (and previous examples), we can obtain phase margins for the circuits in Fig.20.

For the loaded, uncompensated circuit (U2), we get just 17°. This does not imply continuous oscillation, but we might expect significant ringing and

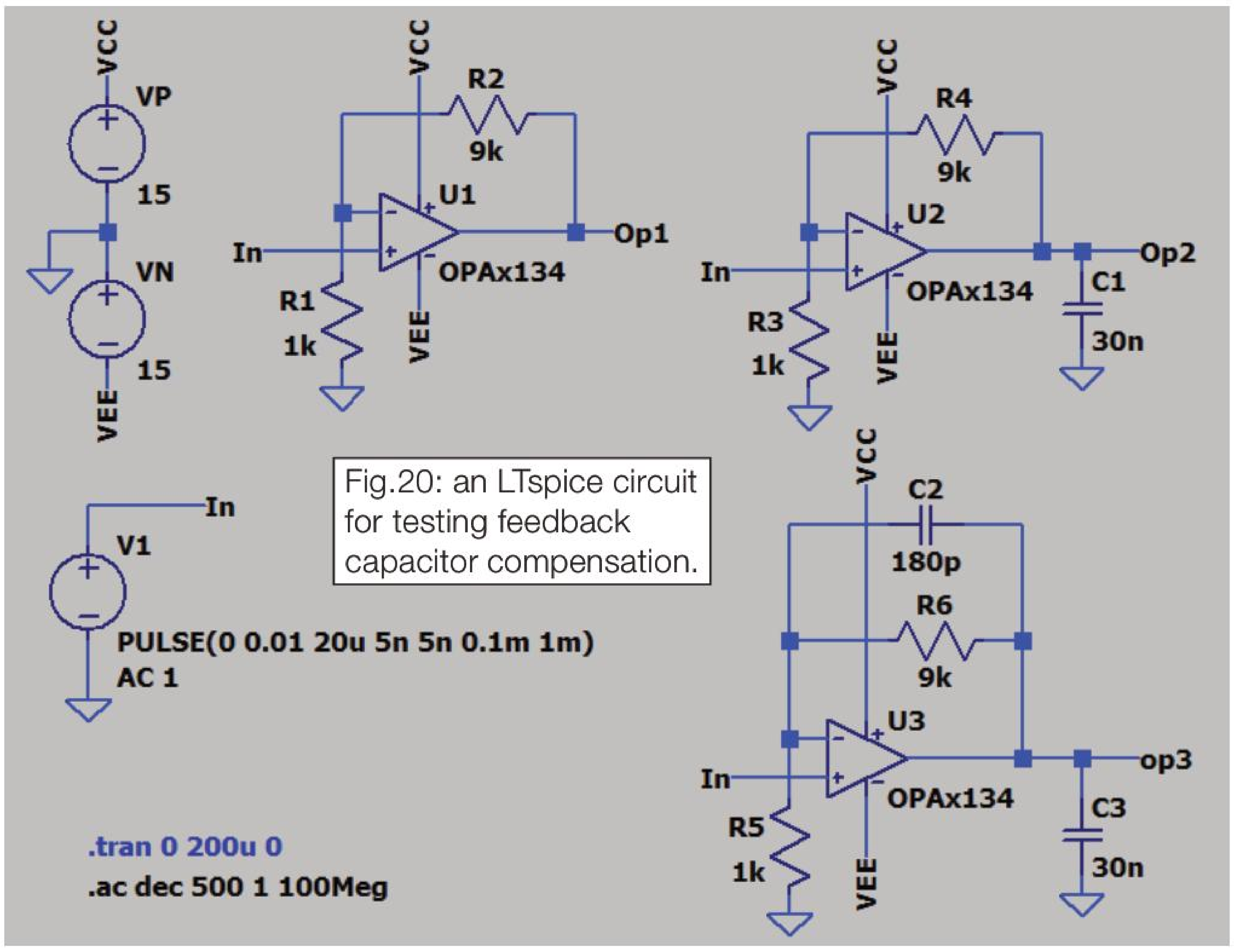


Fig.20: an LTspice circuit for testing feedback capacitor compensation.

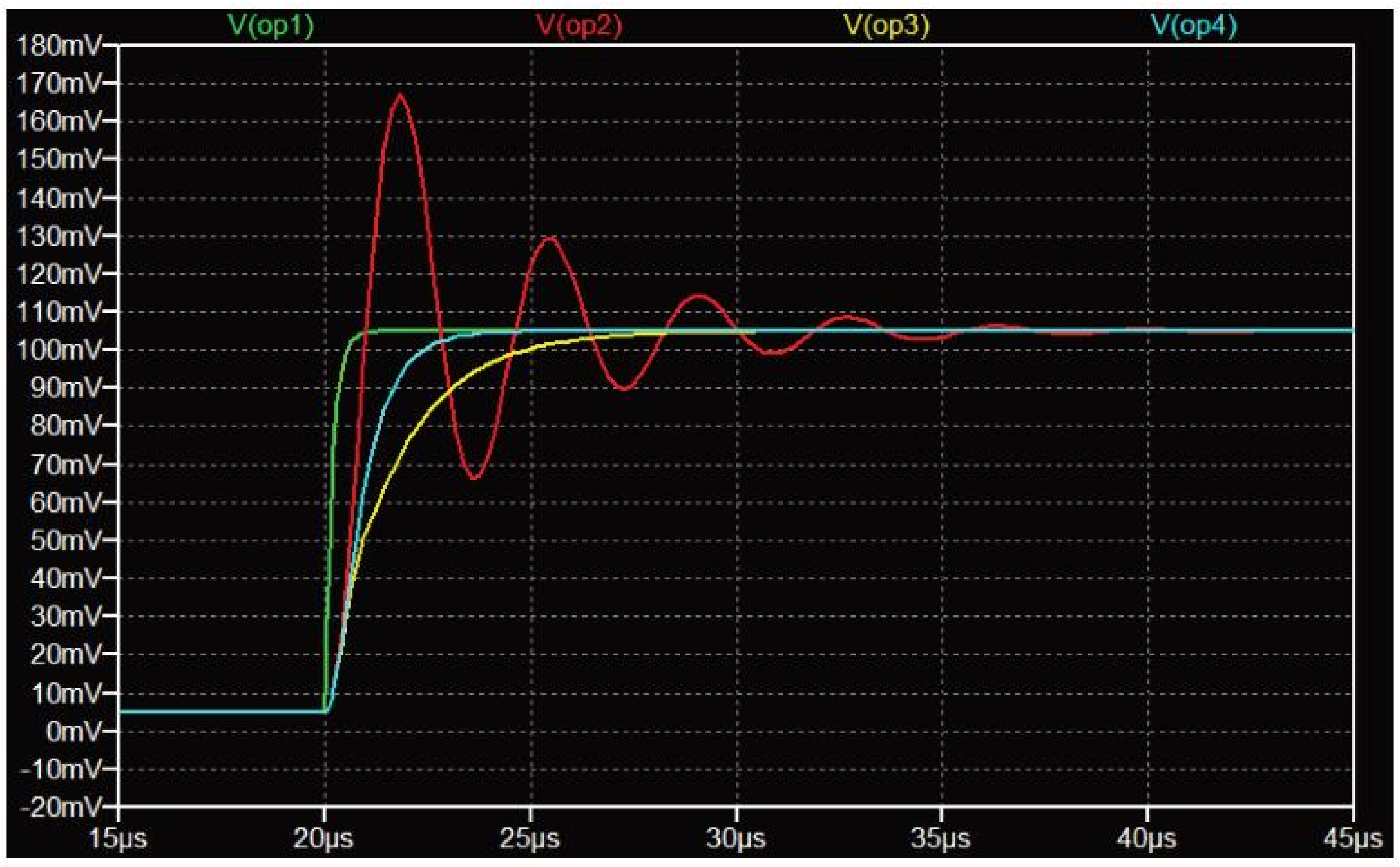


Fig.21: transient analysis (step response) for the circuits in Fig.20.

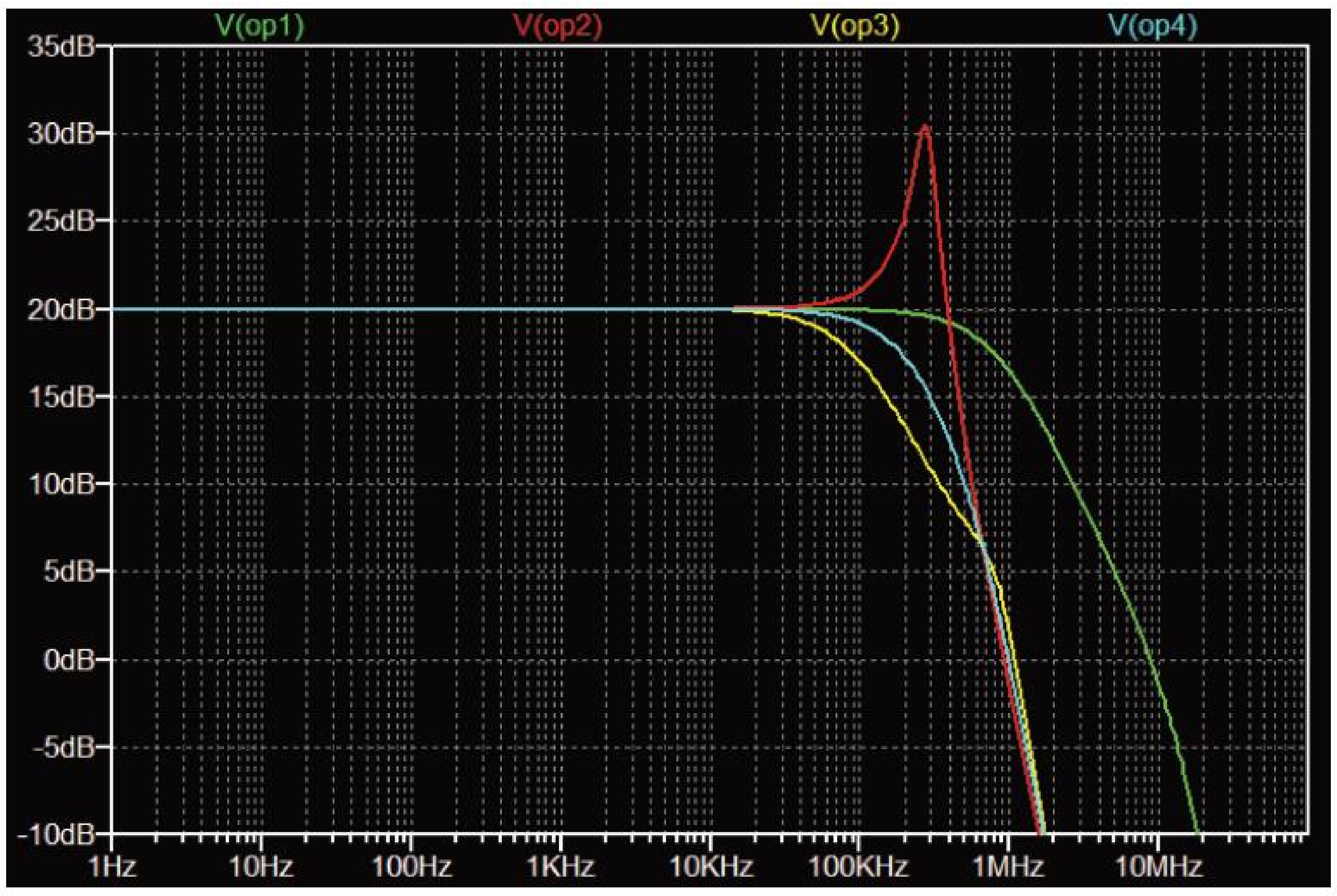


Fig.22: the AC analysis results for the circuits in Fig.20.

the potential for full instability with a shift in conditions. With the feedback capacitor (U3), the phase margin improves significantly to 51°.

Fig.21 shows the step responses from the circuits in Fig.20, with an additional copy of the circuit using U4 with the same values as the U3 circuit, except $C_F$ = 100pF, rather than 180pF as calculated above. The U4 circuit has a better phase margin of 63°.

In Fig.21, we see significant overshoot and ringing for the uncompensated capacitive load (op2, red trace), which is eliminated by the feedback capacitor. Using $C_F$ = 100pF (op4, cyan trace) provides a faster step response than $C_F$ = 180pF (op3, yellow trace), indicating that some experimentation with values can optimise the value compared with the calculation given above.

Similarly, the frequency responses in Fig.22 show significantly peaking for the loaded circuit without compensation (red trace) and improved bandwidth for compensation with $C_F$ = 100pF (cyan trace) compared with $C_F$ = 180pF (yellow trace).

This completes our look at load capacitance compensation for op amps. Unwanted oscillations are a common problem and, although we have not covered every possible approach, we have looked at many of the key techniques for stabilising troublesome op amp circuits.

## Conclusion

We have looked at several commonly used approaches to improving the stability of op amp based amplifier circuits. Not doubt there are other methods that can be used. Which one you choose will depend on the op amp being used, the closed-loop gain, how much capacitance the output is exposed to and other factors.

These methods can be used in combination, if necessary. More phase margin may be necessary for circuitry that feeds connectors going off-board, such as long audio cables. PE

ETI BUNDLE (1) Teach-In 3, 4 and 5 – all on CD-ROM – only £18.95

**ELECTRONICS TEACH-IN 3 – CD-ROM**
**Mike & Richard Tooley**

The three sections of the *Teach-In 3* CD-ROM cover a huge range of subjects that will interest everyone involved in electronics – from newcomers to the hobby and students to experienced constructors and professionals.

The first section (80 pages) is dedicated to *Circuit Surgery*, *EPE/PE's* regular clinic dealing with readers' queries on circuit design problems – from voltage regulation to using SPICE circuit simulation software.

The second section – *Practically Speaking* – covers hands-on aspects of electronics construction. Again, a whole range of subjects, from soldering to avoiding problems with static electricity and identifying components is covered. Finally, our collection of *Ingenuity Unlimited* circuits provides over 40 circuit designs submitted by readers.

The CD-ROM also contains the complete *Electronics Teach-In 1* book, which provides a broad-based introduction to electronics in PDF form, plus interactive quizzes to test your knowledge and TINA circuit simulation software (a limited version – plus a specially written TINA Tutorial).

The *Teach-In 1* series covers everything from electric current through to microprocessors and microcontrollers, and each part includes demonstration circuits to build on breadboards or to simulate on your PC.

**ELECTRONICS TEACH-IN 4 – CD-ROM**
**A BROAD-BASED INTRODUCTION TO ELECTRONICS**
**Mike & Richard Tooley**

The *Teach-In 4* CD-ROM covers three of the most important electronics units that are currently studied in many schools and colleges. These include, Edexcel BTEC level 2 awards and the electronics units of the Diploma in Engineering, Level 2.

The CD-ROM also contains the full *Modern Electronics Manual*, worth £29.95. **The Manual** contains over 800 pages of electronics theory, projects, data, assembly instructions and web links.

A package of exceptional value that will appeal to anyone interested in learning about electronics – hobbyists, students or professionals.

**ELECTRONICS TEACH-IN 5 – CD-ROM**
**JUMP START**
**Mike & Richard Tooley**

15 design and build circuit projects for newcomers or those following courses in school and colleges.

The projects are: ■ Moisture Detector ■ Quiz Machine ■ Battery Voltage Checker ■ Solar-Powered Charger ■ Versatile Theft Alarm ■ Spooky Circuits ■ Frost Alarm ■ Mini Christmas Lights ■ iPod Speaker ■ Logic Probe ■ DC Motor Controller ■ Egg Timer ■ Signal Injector Probe ■ Simple Radio Receiver ■ Temperature Alarm.

**PLUS**

*PIC'n' Mix* – starting out with the popular range of PIC microcontrollers and *Practically Speaking* – tips and techniques for project construction.

The CD-ROM also contains:
■ Complete *Teach-In 2* book, a practical introduction to PIC microprocessors
■ MikroElektronika, Microchip and L-Tek PoScope software.

Three Teach-ins for the great price of £18.95

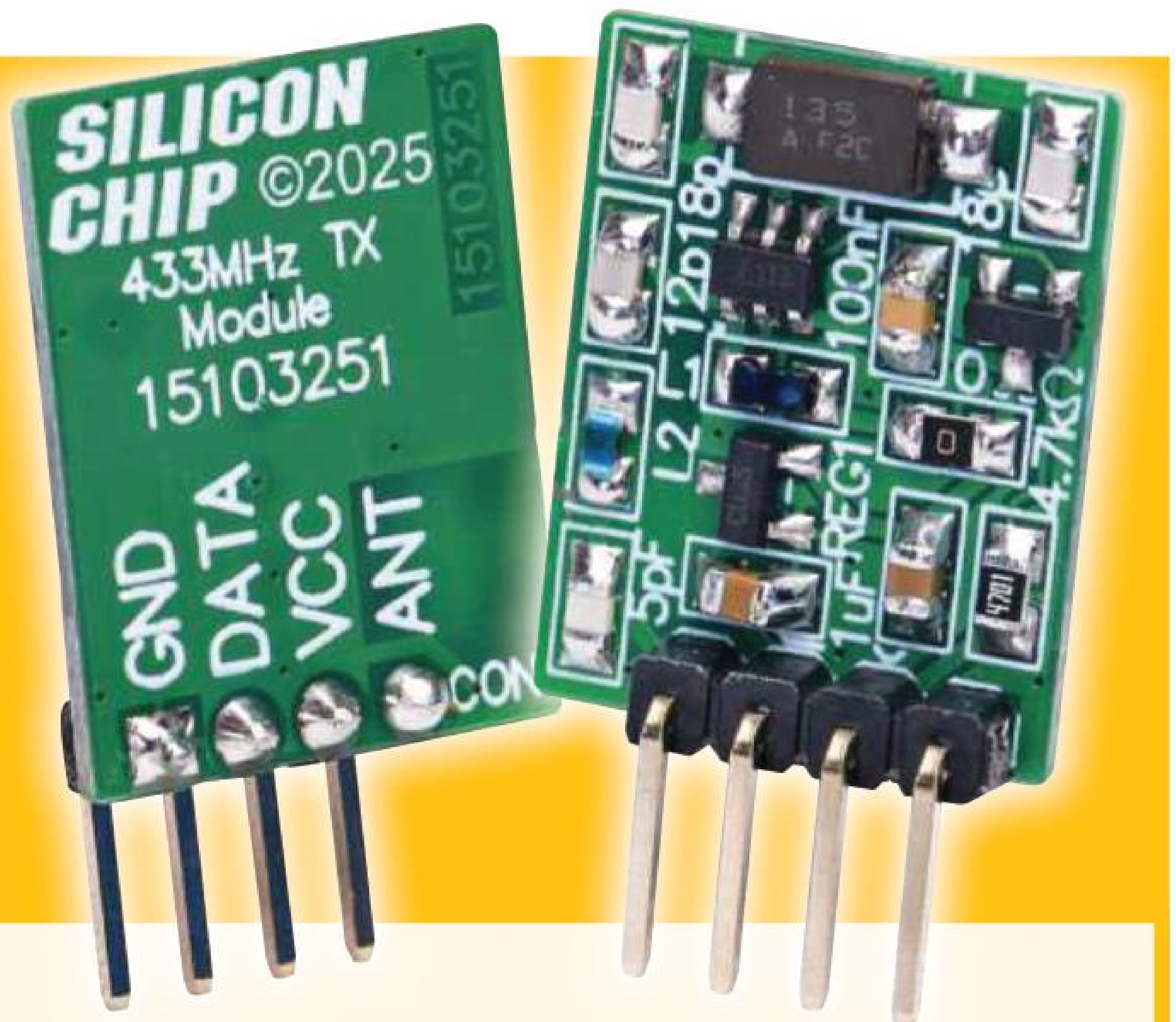
PLUS you also get the contents of the free CD-ROM from each issue...

... so that's another TWO Teach-Ins and The Full Modern Electronics Manual!

What a Bargain!!

PROJECT BY TIM BLYTHMAN

# 433MHz Transmitter Module



**The 433-434MHz LPD433 (low interference potential device) band is used by many devices for radio control. There is no need for a class licence, so low-cost 433MHz modules make it easy to create a wireless link. As it can be difficult to get these modules with a consistent interface and good performance, we have created our own version, which is a direct replacement for commercial equivalents.**

**We've published many projects** that operate on the 433MHz LIPD band. Many include a transmitter and receiver module pair, or one or the other to interface with existing equipment. The existence of drop-in radio modules has made this quite easy.

In the UK, regulation of radio spectrum is handled by Ofcom.

Licence-exempt operation of low-power devices in the 433MHz band is permitted under Ofcom's Short Range Device (SRD) framework, as defined in its Interface Requirement documents (notably IR2030), which align with European CEPT and ETSI standards.

The SRD allocation covers the frequency range from 433.05MHz to 434.79MHz, meaning that the band is actually centred closer to 434MHz than 433MHz. Many devices operate at the commonly used nominal frequency of 433.92MHz.

Under these provisions, no individual radio licence is required, provided the device complies with the prescribed technical conditions. These include limits on effective radiated power, duty cycle, modulation type, occupied bandwidth and spurious emissions, as specified in the relevant ETSI standards referenced by Ofcom.

Typical licence-exempt devices in this band operate at low transmit power (often around 10mW EIRP, depending on device category).

SRD operation in the UK is on a non-protected basis. Devices are not guaranteed freedom from interference and must tolerate it if it occurs. The 433MHz band is shared with other services, including licensed amateur radio users operating in the adjacent 70cm band, who may legally transmit at much higher power levels.

In practice, this means that devices such as garage door remotes, wireless sensors and key fobs generally coexist without problems, but interference is possible, particularly in built-up areas or where many devices share the same frequency.

Provided the relevant technical limits are met, anyone may operate a compliant transmitter in this band in the UK without applying for or paying for a radio licence.

## Past projects

Nearly all of our projects using 433/434MHz transmitters and receivers have used prebuilt modules. However, we recently started incorporating discrete transmitter circuits into some of our designs. The Remote Control Range Extender from January 2023 and the Secure Remote Switch from December 2024 and January 2025 both did so.

They incorporate custom circuitry built around an RF chip, the MICRF113 ASK Transmitter IC. ASK is short for 'amplitude-shift keying', where digital data is encoded as different amplitude levels of the carrier wave.

To ensure all these projects could still be built, we decided to design this drop-in replacement for the tiny transmitter modules. Our Transmitter design has the same size, shape and pinout, so is interchangeable. Later we'll look at some minor differences from the modules; they are mostly improvements made possible by the MICRF113 chip.

## Module overview

We also thought that this was a

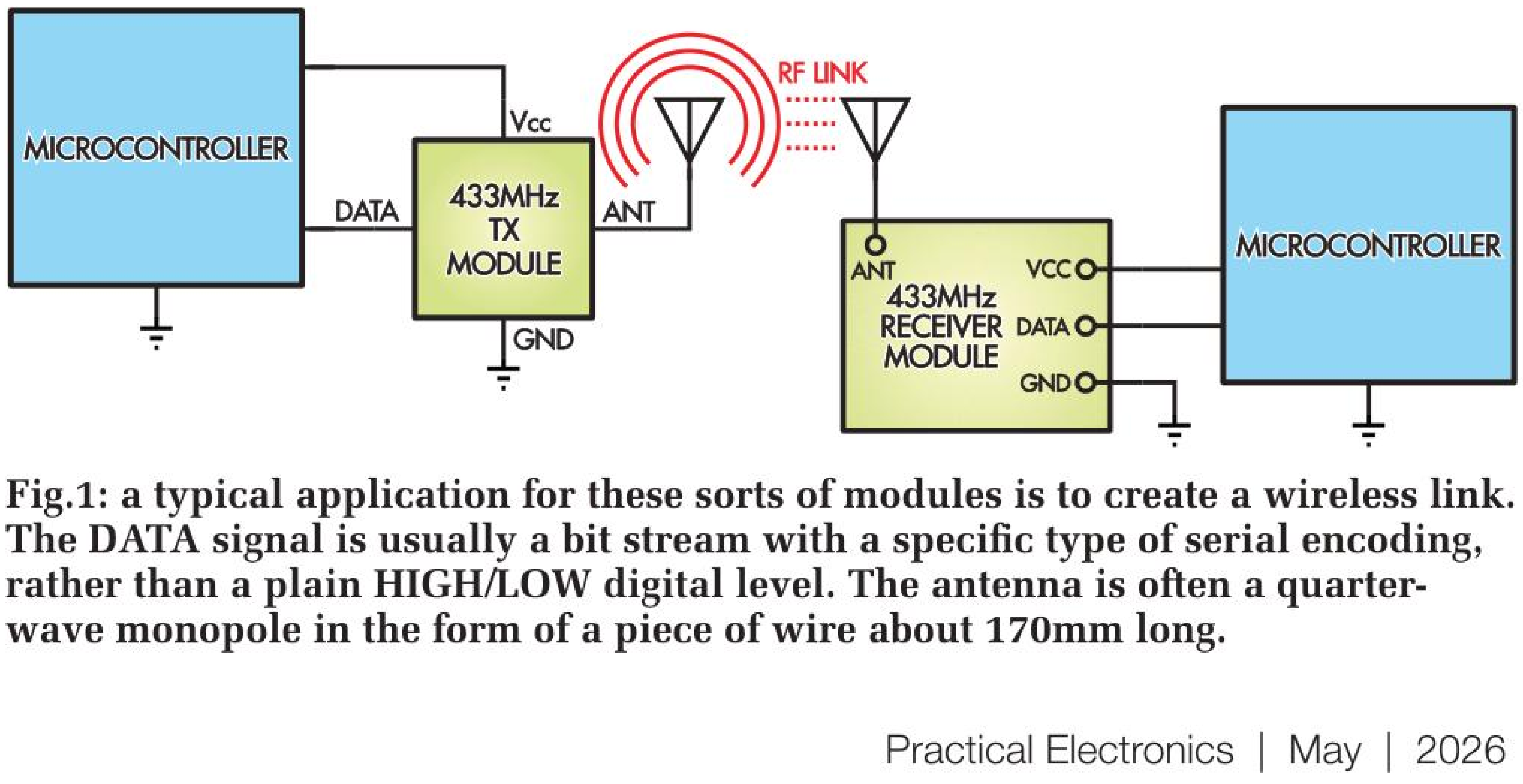


**Fig.1: a typical application for these sorts of modules is to create a wireless link. The DATA signal is usually a bit stream with a specific type of serial encoding, rather than a plain HIGH/LOW digital level. The antenna is often a quarter-wave monopole in the form of a piece of wire about 170mm long.**

# 433.9MHz Transmitter Module

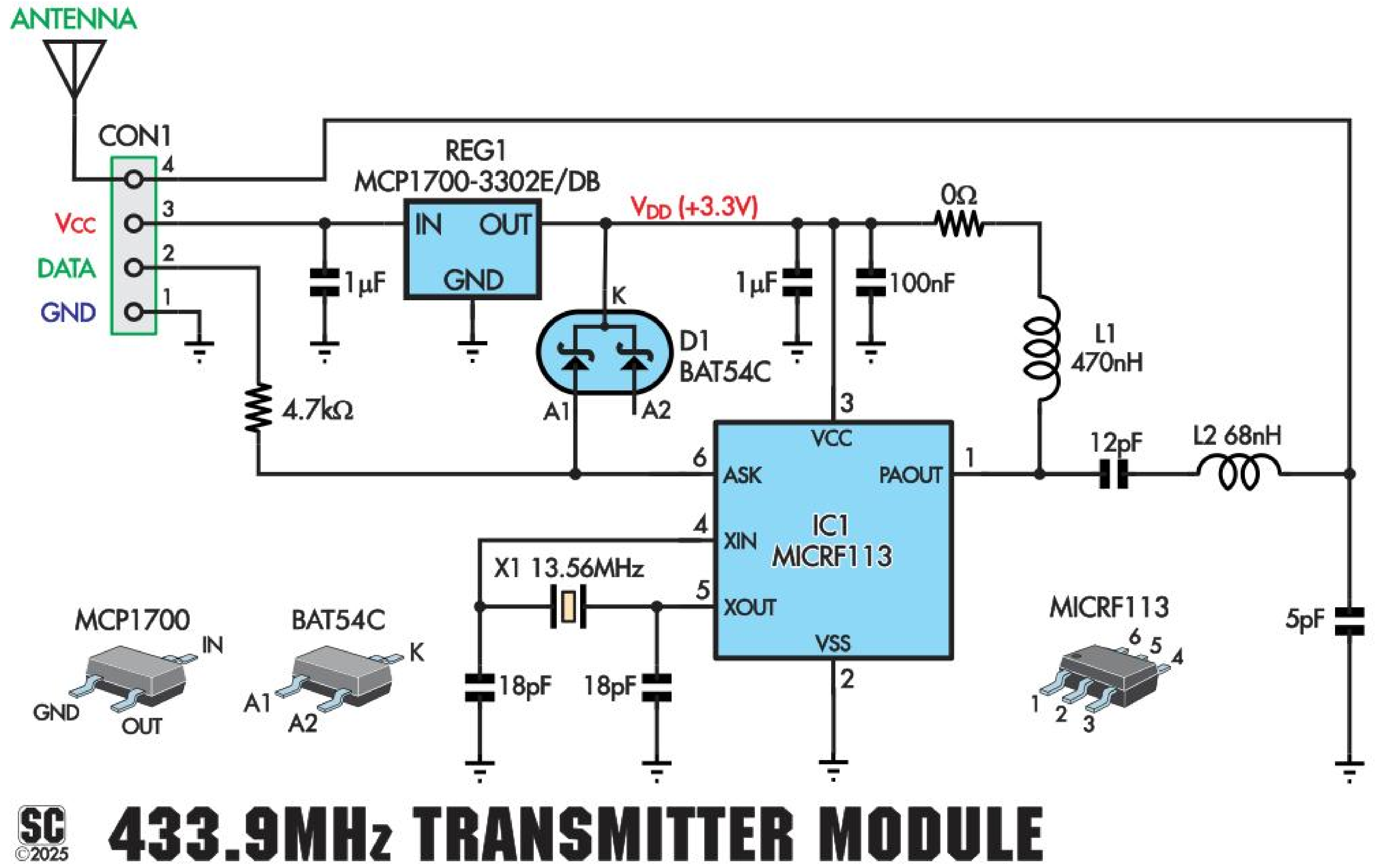


**Fig.2: the Transmitter circuit mostly consists of the recommended RF capacitors and inductors for the MICRF113, along with a 13.56MHz crystal to set the carrier frequency. We added a voltage regulator and DATA input protection circuitry to allow operation with 3.3V or 5V systems.**

good time to take a deeper look at these modules, their general operating parameters and what their limitations are. There are of course many cheap versions available from online sellers, and the available information can be limited.

The transmitter module is our focus in this article, although we will look at how the receiver extracts the signal of interest from the busy radio spectrum. This information is presented at the end of the article, in case you are only interested in building the Transmitter.

Fig.1 shows a typical arrangement used to provide a wireless link. It is as simple as supplying power to the transmitter module and providing a data signal. What goes in the DATA pin on the transmitter generally comes out at the DATA pin on the receiver, subject to the limitations of the modules, which we also discuss later.

Transmission over distances of up to 100m are possible in open air.

Usually, the data is an encoded packet reporting a status or a command. The packets are brief (perhaps 0.1s in duration) and are only transmitted occasionally, which helps to avoid interference with other equipment. Even if one packet interferes with another device, it's unlikely the intervals between packets will match exactly, so they will only interfere occasionally.

The software at each end is often designed to encode an identity (possibly using DIP switches or the like), as well as the command or data. This is so that a similar nearby transmitter does not trigger a receiver unless both share the same identity value. Some codes also include a checksum to validate the integrity of the data.

## Circuit details

Much of the circuitry needed, shown in Fig.2, is outlined in the data sheet for the MICRF113 chip. It should come as no surprise that it is also similar to the Discrete Transmitter for the Secure Remote Switch.

As well as IC1, the MICRF113, there is 13.56MHz crystal X1, along with its requisite load capacitors, operating as a frequency reference. IC1 uses an internal ×32 PLL (phase-locked loop) to multiply that up to 433.92MHz. The capacitors and inductors on the PAOUT pin (which delivers the RF signal) form a signal matching network to limit the transmitted harmonics.

The LINK resistor should normally be a 0Ω jumper, but a resistor can be used instead to reduce the output RF power and thus the necessary supply current. There are more details on this in the MICRF113 data sheet. A solder blob across its pads would also work!

The MICRF113 has an operating supply voltage range of 1.8-3.6V; we have chosen a nominal 3.3V and added a voltage regulator to provide this. The current demand is about 2mA in the quiescent state and 12mA when transmitting, so a small regulator is all that is needed.

We used the MCP1700, which can handle up to 6V at its input, as we expect in most cases it will be receiving 5V. Its dropout voltage at the expected load levels is less than 0.1V, so it won't cause a problem if a lower supply voltage like 3.3V is used instead. The two 1µF capacitors are required by the regulator for stability, while the 100nF capacitor provides local supply bypassing for IC1.

Finally, the incoming data signal (from a microcontroller or the like) comes through a series resistor with a shunt diode to limit the voltage at the ASK pin. Thus, our Transmitter is compatible with 5V and 3.3V supply and logic levels.

## Assembly

The Transmitter has been designed to fit in the same compact footprint as similar commercial modules, which tend to have similar pinouts. To keep this device a similar size to commercial offerings, the assembly involves working with small SMD parts.

The MICRF113 chip comes in the SOT-23-6 package, while most of the passives are M2012 (0805 imperial) at 2 × 1.2mm, with one slightly smaller (M1608/0603 or 1.6 × 0.8mm) part.

You'll need the standard SMD gear such as flux paste, tweezers and so forth. A magnifier and good light will be a great help with such small parts. Fume extraction is always recommended when working with flux paste, and you should have a solvent to clean up the flux residue, too.

We've managed to fit all the parts on one side, so you can use some Blu-Tack or similar to keep small the PCB in position on your workbench. Solder wicking braid will be handy to remove any solder bridges that might occur.

We've managed to fit practically all the component designators on the PCB silkscreen, but it's best to refer to the overlay diagram (Fig.3) and accompanying photo to confirm the location of the components.

Start by spreading a thin layer of flux paste on all the PCB pads. IC1 has the smallest pin pitch, so place it first. Its pin 1 designator is very small, so you will probably need to examine it under magnification to confirm the correct orientation. It must be aligned

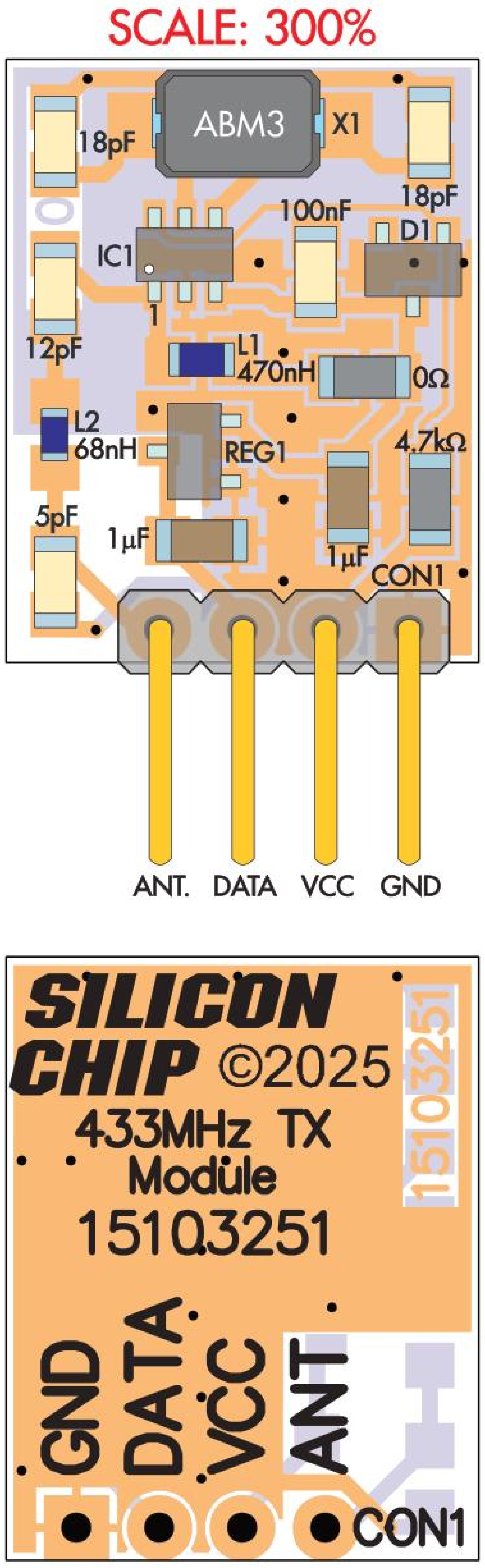


Fig.3: the PCB is quite small, but we managed to fit most of the component designators on the PCB silkscreen. You can check the components on the overlay diagram here as you go.

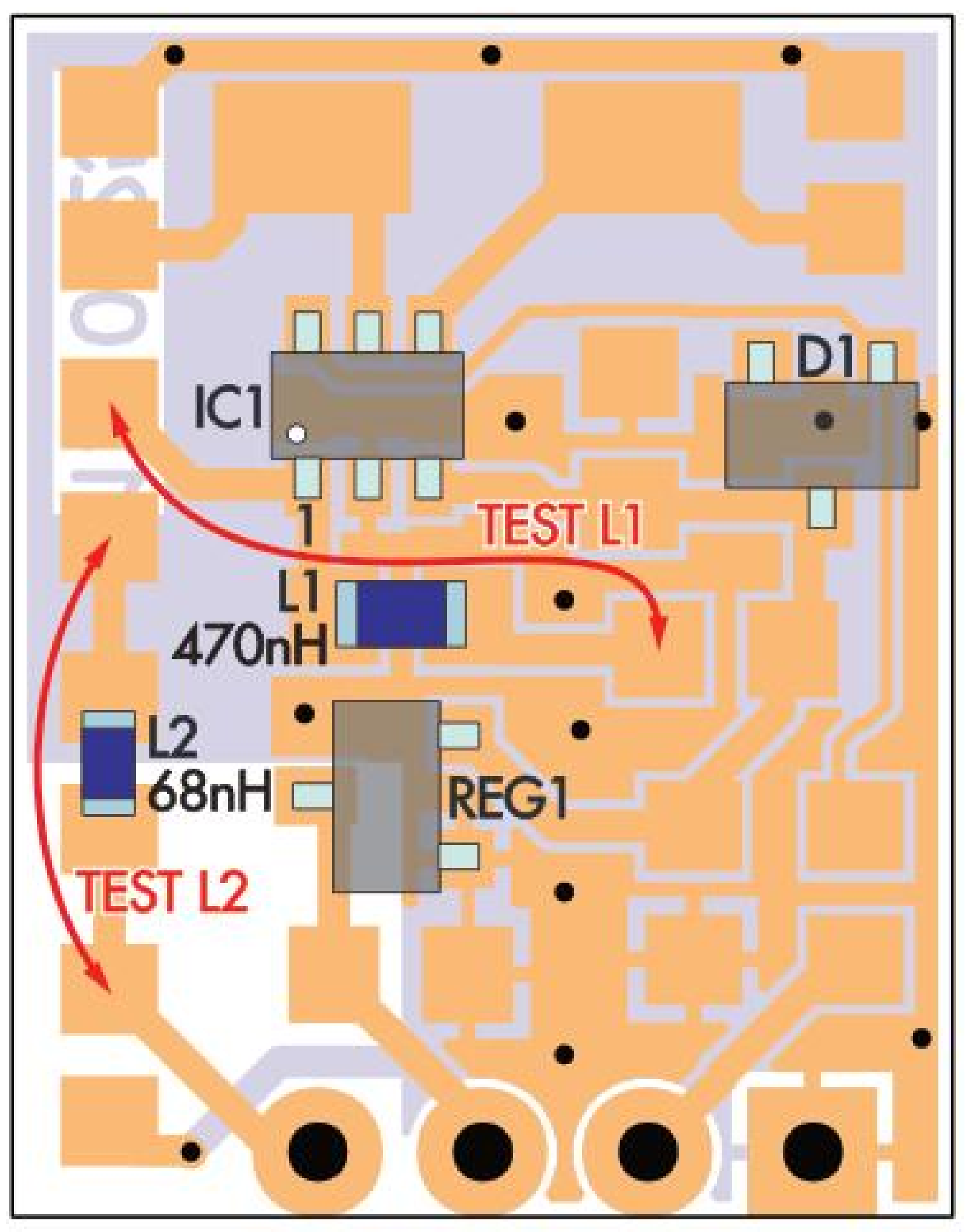
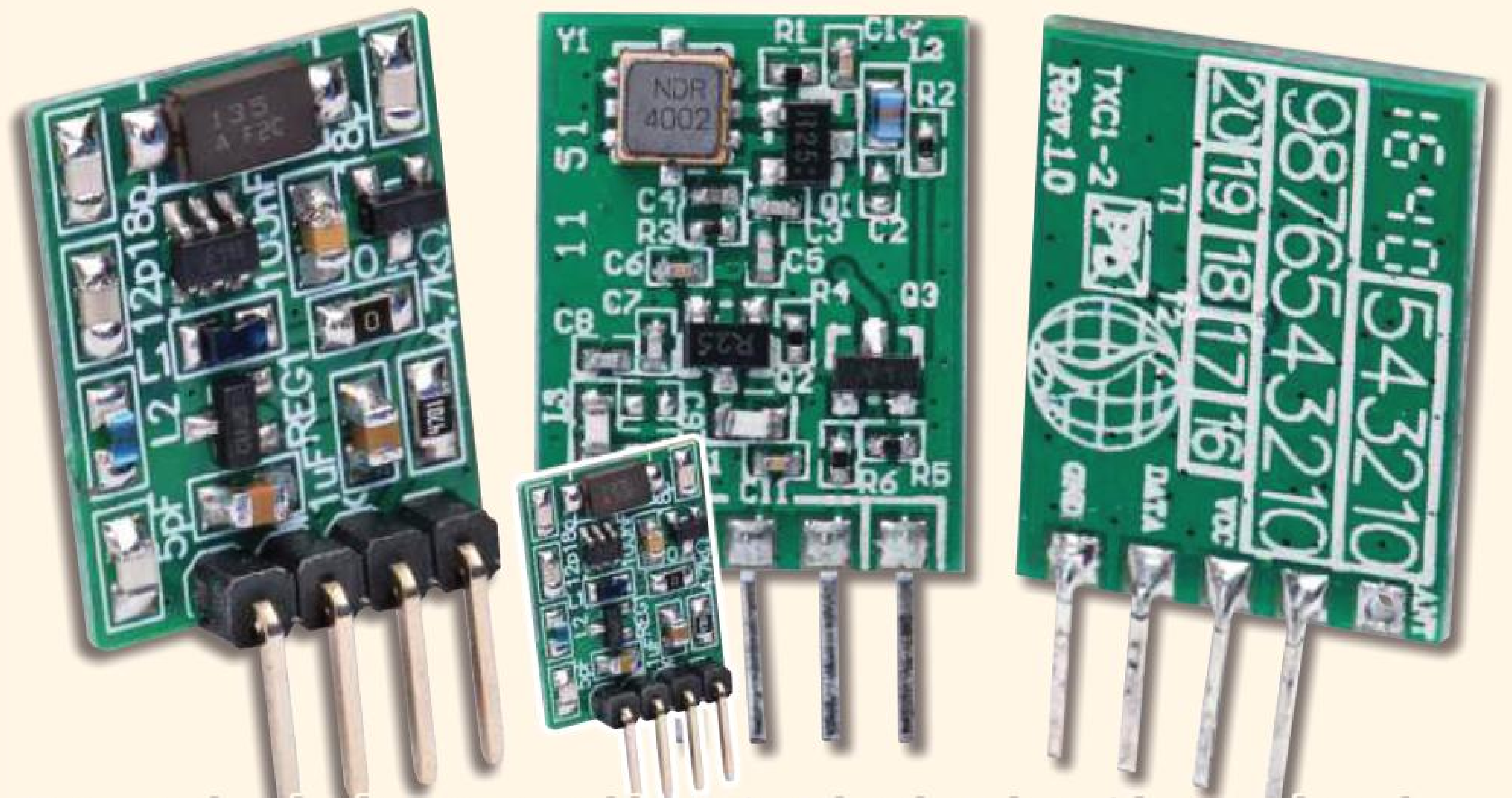


Fig.4: with the components fitted as shown here, you can probe for continuity at the marked pads, which will indicate whether the inductors have been soldered correctly.



You can also check your assembly against the photo here (shown enlarged and at actual size on the left). We used a right-angled header, but you might like to use straight headers to allow the module to be mounted parallel to a PCB, like we did with the Secure Remote Switch. We've designed our Transmitter to be a drop-in replacement for the prebuilt module shown in the centre and on the right, which is a fairly common type. It is even the same size with the same pinout.

with the matching mark on the PCB silkscreen, which is near L1.

If you have the PCB upright, with X1 at the top and the external connections at the bottom, the text on IC1 should be upright, too.

Clean the tip of your iron and add some fresh solder. Tack one lead and confirm that the others are flat and within their pads. If needed, adjust the part position by using the iron to remelt the solder. Then solder the remaining leads.

Use solder-wicking braid to remove any solder bridges as you go. It will be trickier as more components are added. Add a little more flux paste, press the braid against the bridge with the iron and then slowly draw both away when the excess solder is taken up.

Next, fit the two SOT-23-3 parts. D1 is near IC1, while REG1 is near the external connector. Fortunately, they should both only fit one way. You can use the same idea; tack one lead, adjust and then solder the remaining leads. These parts have leads that are quite spread out, so they should not bridge easily.

Next, fit the two inductors. These will be a bit fiddly, since their leads are only on the undersides. You will need to apply the iron to the PCB pad and add solder, allowing it to melt and flow onto the leads. The smaller 68nH part is L2, which sits between the 5pF and 12pF capacitors, while 470nH inductor L1 is between IC1 and REG1.

While L2 is the smaller M1608/0603-sized part, we have used M2012/0805 pads to make soldering it easier. All the remaining two-lead parts are in M2012/0805 packages.

If you are unsure that the parts have been soldered correctly, you can check this with a multimeter set to continuity or resistance modes. The inductors have low DC resistance, so both should read well under 10Ω. With none of the surrounding parts fitted, they are safe to probe.

For L2, probe the adjacent pads on the 12pF and 5pF capacitors, as shown in Fig.4. Using a nearby pad eliminates the chance of a false positive in the event that the component is connected to the solder in the joint but not the pad below.

For L1, try the other end of the 12pF capacitor and the end of the 0Ω link next to L1. If you get a low resistance reading across each inductor then all is well. If not, try adding some flux paste to each joint and reflow the solder with your iron before checking again.

## The passives

Fit the crystal, X1, next; it is unpolarised, as are the other remaining parts. Its leads are quite small, so you might need to use a similar soldering technique to the inductors. The larger PCB pads should make this easy, although you won't be able to test for continuity in the same fashion.

Be careful not to mix up the capacitors. The two 18pF capacitors are adjacent to the crystal and then, on the left of the PCB, are the 12pF capacitor above L2 and the 5pF crystal below it. The two 1µF capacitors are near REG1, while the 100nF part is next to IC1.

Move on to install the 4.7kΩ resistor in the bottom-right corner and the 0Ω link nearby. Finally, fit the header of your choice; we used a right-angle header to match the prebuilt transmitters.

Use a solvent to thoroughly clean the flux from the PCB and allow it to dry. Inspect your soldering with a magnifier and confirm that all the components are soldered correctly with no bridges.

### Testing

If you wish to proceed cautiously, you should power up the Transmitter from a current-limited supply. The Transmitter should draw around 3mA while idle or 15mA when transmitting. Something simple, like a 330Ω resistor in series with a 5V supply, would also be suitable.

Add a jumper wire or similar between the GND and DATA pins to ensure that the Transmitter is initially in the idle state. Then apply power and measure voltage across the resistor; it should be no more than around 1V.

The next step is to apply a waveform to the DATA pin and see that it is received correctly. You may have a project planned for the Transmitter, in which case you should simply connect it and try it out.

Another simple test we tried can be done with a piezo transducer and a 433MHz receiver module. Wire up the receiver module so that the piezo is connected between its DATA output and GND, then connect 5V power. You might not need an antenna to test over short distances, such as on a workbench.

The piezo should emit a sound like white noise or static; this indicates that it is picking up normal background RF noise. If you then power on the Transmitter and drive its DATA input high, the noise should cease as that signal overwhelms the background noise and saturates the receiver's automatic gain control (AGC).

Driving the DATA pin low should similarly cause the background noise to resume. Applying a 1kHz square wave to the DATA pin should cause a high-pitched noise to be emitted from the piezo. Note that you should always make sure the DATA pin is driven, since it is a high-impedance input and could otherwise float to an unknown level.

### Conclusion

You can read more about some of our comparative tests in the text below, but we have found our Transmitter to be just as good, if not better than, other similar transmitter modules it is a drop-in replacement for.

It is capable of transmitting at 10mW, which could fall foul of the EIRP restrictions if used with a highly directional antenna. So we recommend sticking to simple antenna designs, such as a quarter-wave dipole, (~170mm for 433MHz) to ensure that you do not exceed the licence limits.

If you must use a directional antenna, replace the 0Ω link with a resistor to reduce the output power (refer to the MICRF113 data sheet).

## A detailed analysis of 433MHz modules

We performed some testing on these modules to ensure our replacement performed at least as well. We purchased a transmitter and receiver module pair from a local electronics shop; these will be our test subjects, alongside our new Transmitter.

The information on their website indicates a maximum supply current of 10mA and a maximum output power of 3dBm (or 2mW) for the transmitter. The suggested data rate is 300bps to 10kbps.

While the 2mW might sound like quite a low transmission power level, the EIRP (equivalent isotropically radiated power) could be higher. It is calculated as though the maximum signal strength (which might only occur in one direction) was radiated in all directions. In fact, the EIRP can be no less than the actual power.

For a theoretical isotropic (outputting the same power in all directions) antenna, the figures will be the same. A highly directional antenna will have higher EIRP since the radiation is concentrated.

Fortunately, we nearly always use non-directional antennas with these modules. The typical gain of quarter-wave monopole antennas is no more than 3dBi, which is about a factor of 2, keeping the EIRP to around 4mW, well under the 10mW limit.

The MICRF113 in the Transmitter is capable of more power, but it too is unlikely to fall foul of the restrictions with a quarter-wave monopole antenna.

The current draw on the MICRF113 peaks at around 13mA during transmission, so it does draw slightly more current for a substantially higher output power.

### Receiver operation

If you have ever monitored the output of the receiver modules while nothing is transmitting, you might have

## Parts List – 433MHz Transmitter Module

1 double-sided PCB coded 15103251 measuring 19 × 15mm
1 4-way pin header, straight or right-angled to suit application (CON1)
1 13.56MHz 5.0 × 3.2mm surface-mounting crystal (X1) **[Abracon ABM3-13.560MHZ-B2-T]**
1 470nH inductor, M2012/0805 size, >434MHz SRF (L1) **[Coilcraft 0805HT-R47TJLB or Murata LQW21HNR47J00L]**
1 68nH inductor, M1608/0603 size, >434MHz SRF (L2) **[Bourns CW160808-68NJ, Coilcraft 0603CS-68NXJLU or Murata LQW18AS68NJ00D]**
1 4.7kΩ M2012/0805 size SMD resistor, ⅛W
1 0Ω M2012/0805 size SMD resistor or value to suit (see text)

**Semiconductors**
1 MICRF113YM6 SOT-23 ASK transmitter IC, SOT-23-6 (IC1)
1 BAT54C/BAT54S/BAT54 200mA 25V schottky diode, SOT-23 (D1)
1 MCP1700-3302 3.3V LDO voltage regulator, SOT-23 (REG1)

**Capacitors** *(all SMD M2012/0805 size, 50V ceramic unless noted)*
2 1µF 50V X7R
1 100nF 50V X7R
2 18pF C0G/NP0 (or to suit crystal)
1 12pF C0G/NP0
1 5pF C0G/NP0

**SC7430 Kit (~£12 + postage):**
*includes all the parts listed here*

noticed the signal is just noise; there is no always low or high idle state. Scope 1 shows a scope grab of the receiver output when no transmission is occurring.

This is due to the way that the receivers resolve signals with different strengths. During transmission, they must be able to deliver a valid signal, whether the transmitter is nearby or far away. In other words, they must be able to cope with receiving weak or strong signals and produce the same output.

The operation of many infrared (IR) receivers is much the same too, although most IR receivers have extra circuitry to suppress the output noise during idle periods.

Though the modules are described as using ASK modulation, it is typically the most simple form called OOK (on-off keying). With OOK, one of the levels is ‘carrier on’ and the other is ‘carrier off’. Here, the carrier refers to the underlying 434MHz signal.

The raw data is also further encoded with the likes of pulse-length, pulse-width or pulse-distance modulation (as also used in IR remote controls). Manchester encoding is another system that is also used in these scenarios. Fig.5 shows representative waveforms of some of these encodings.

The simple on-off nature of the data means that the receiver only needs to recognise two signal levels, and this is done with the assistance of automatic gain control (AGC). This is much the same principle that ensures that nearby and far AM radio stations are received at the same volume.

The AGC takes an average of the incoming signal strength, compares it to the instantaneous strength and adjusts the receiver gain to compensate. The AGC response time will also dictate a minimum data rate; if the receiver sees a carrier on state for too long, it will saturate and start producing noise.

This is why the various encoding schemes have a duty cycle close to 50%. It means that the carrier on and carrier off levels are a similar distance from the average that the AGC sees, and both output levels are decoded correctly.

## Analog behaviour

There are some reports of these modules being capable of transmitting and receiving analog data, such as voice

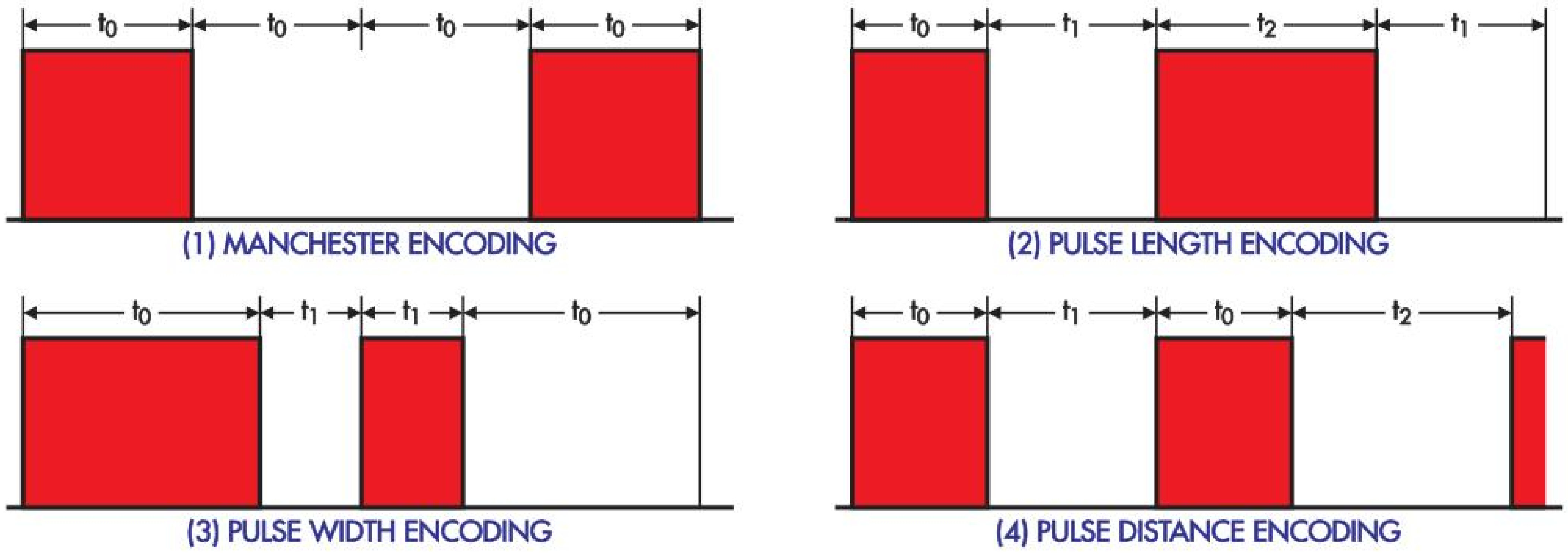
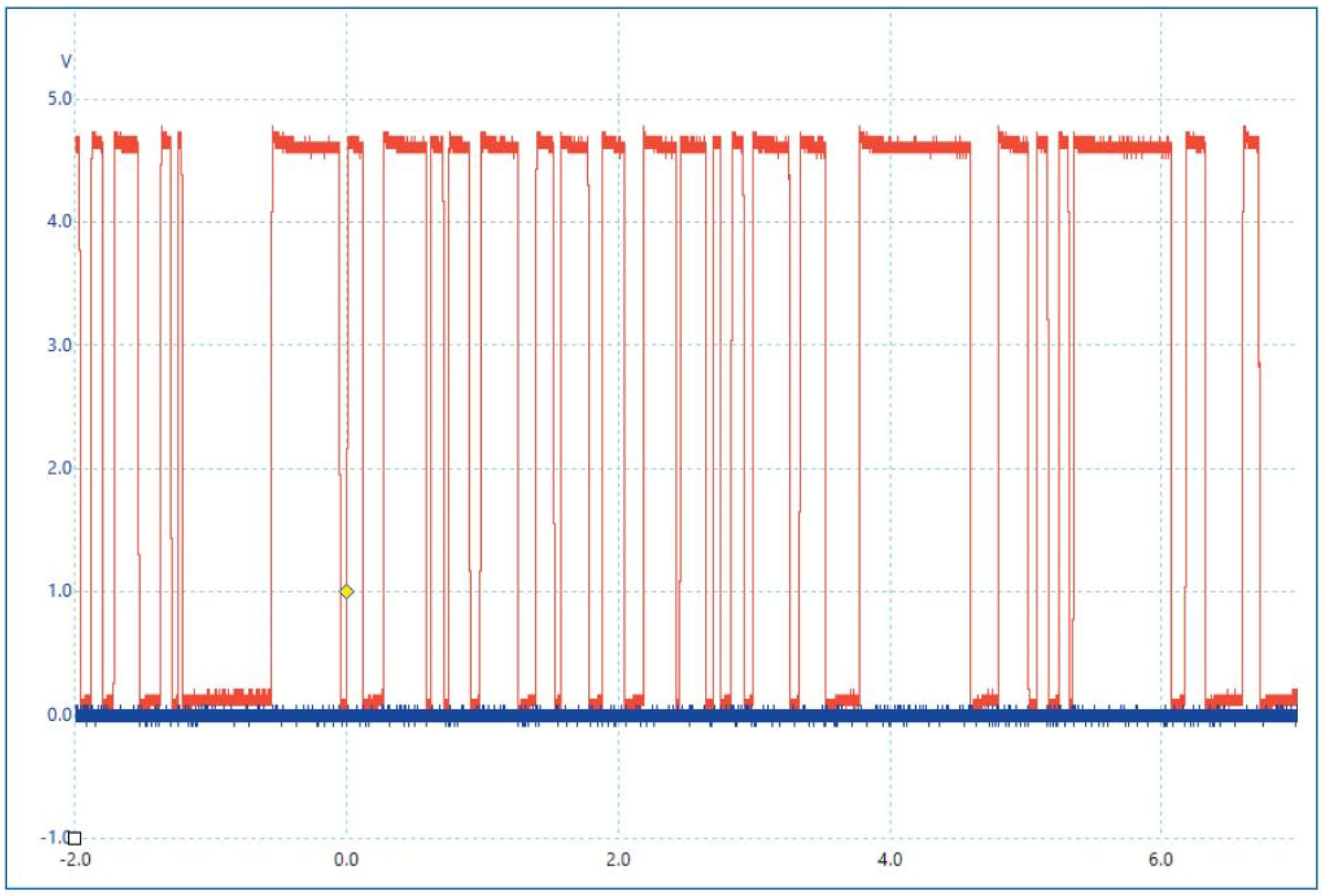
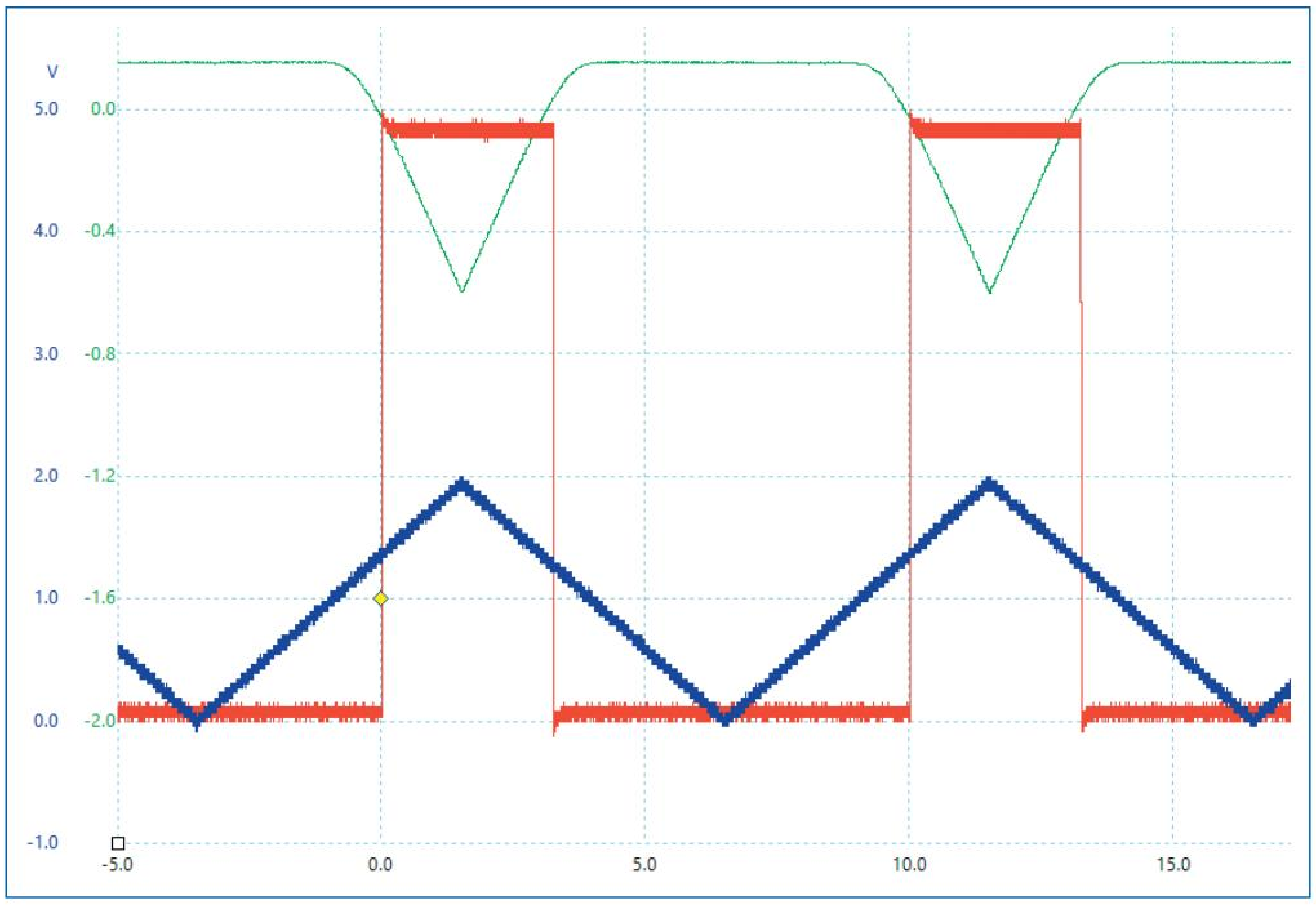


**Fig.5: examples of various encodings that are used with RF (and IR) systems. The pulses are different to encode a 0 or 1 while maintaining a duty cycle near 50%. Manchester encoding is often decoded by looking for the rising or falling edge in the middle of the bit time, rather than the pulse length or spacing.**



**Scope 1: in the absence of a strong, nearby transmitter, the receiver modules will deliver noise. When connected to a piezo transducer or similar, it sounds like white noise.**



**Scope 2: the current draw (and thus output power) of the commercial transmitter module shows an analog response to a triangle wave, suggesting it is capable of amplitude modulation (AM) to some extent.**

or audio, using AM (amplitude modulation). But it's doesn't appear to be possible with either the commercial receiver or our Transmitter.

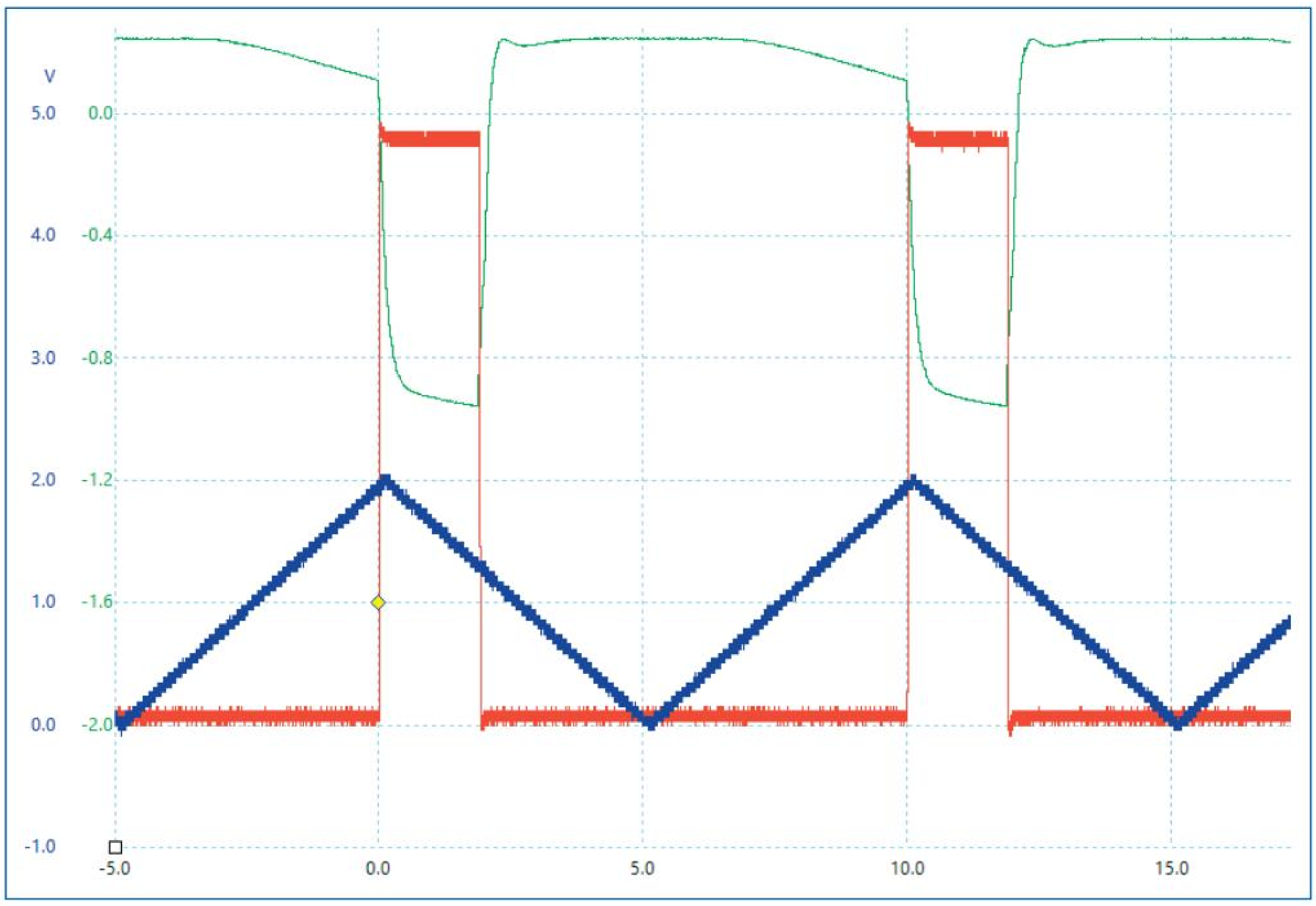
The commercial transmitter might be capable of AM transmission, so could be used for this purpose with an appropriate receiver. Scope grabs of our tests are shown in Scope 2 (commercial Transmitter) and Scope 3 (our Transmitter).

In these, the blue trace is a triangle waveform from a signal generator, which was fed into the DATA input of the transmitter module. The red trace is the output of a nearby receiver, and the green trace is the voltage across a resistor in the transmitter's positive supply; a crude current measuring shunt.

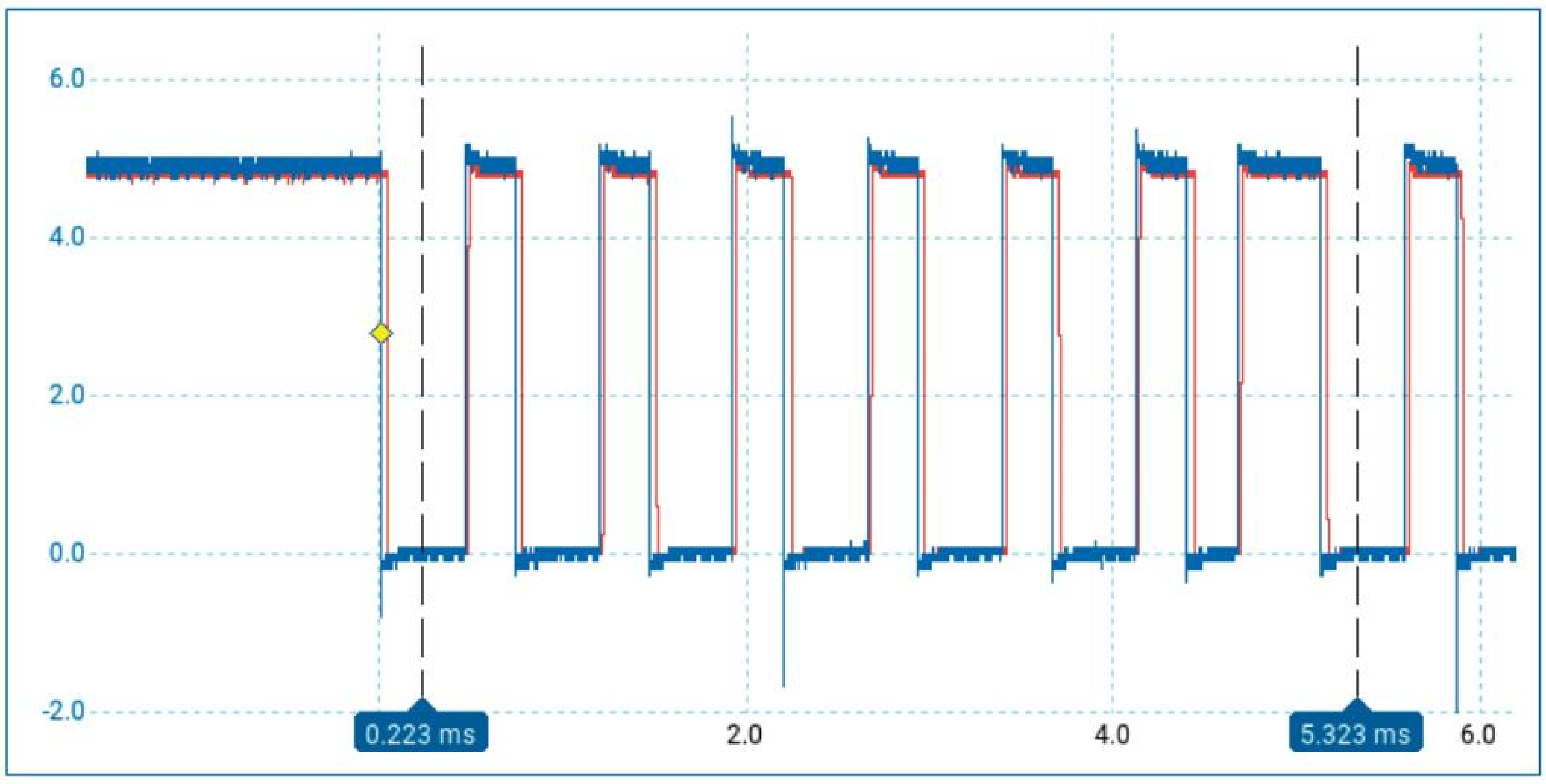
Being in the positive supply, a lower voltage means more current being consumed by the transmitter. Assuming that the current reflects the strength of the RF transmission, we can get an idea of whether the modulation is AM or simple OOK.

For Scope 2 (the commercial transmitter), we can see that the current does indeed follow the incoming signal level over a range, while Scope 3 (our Transmitter) shows a very digital response, with a hysteresis between 1.3V (falling) and 2.0V (rising). So we don't think our module will be suitable for AM transmission.

In both cases, you can see that the receiver has a very 'digital' response, so we don't think it could be used for AM reception. There is what appears to be some hysteresis in the receiver output, but part of that may be delays in the receiver and its AGC response.



**Scope 3: our Transmitter has a more digital response, even showing hysteresis. This should provide a cleaner signal when used in digital applications, as is usually the case for these modules.**



**Scope 4: the blue trace shows a transmitter input, while the red trace is a receiver output. The short delay between the two is around 20-40μs and differs slightly between the modules.**

## Comparative tests

To keep the tests between our module and the commercial one fair, we set everything up on a breadboard so that we could swap between the two transmitters without changing anything else.

We did not attach any external antenna to the modules. The breadboard strips provide a very short antenna that was sufficient for transmission over short ranges.

We looked at aspects like signal delay between the transmitter and receiver and the response to transmitting at different data rates. For all these, we used our Arduino Uno to generate a pulse-width waveform of the type used in 433MHz applications. Each cycle is nominally 720μs, giving a 1.4kbit/s data rate.

We used an oscilloscope to compare the data coming from the Uno to the data received by the receiver module.

Scope 4 shows the delay between the input to the transmitter and the output from the receiver. For both transmitters, the rising edge was propagated more quickly than the falling edge (by about 10μs). Also, our Transmitter showed quicker response times in general; around 20μs compared to 35μs for the commercial module.

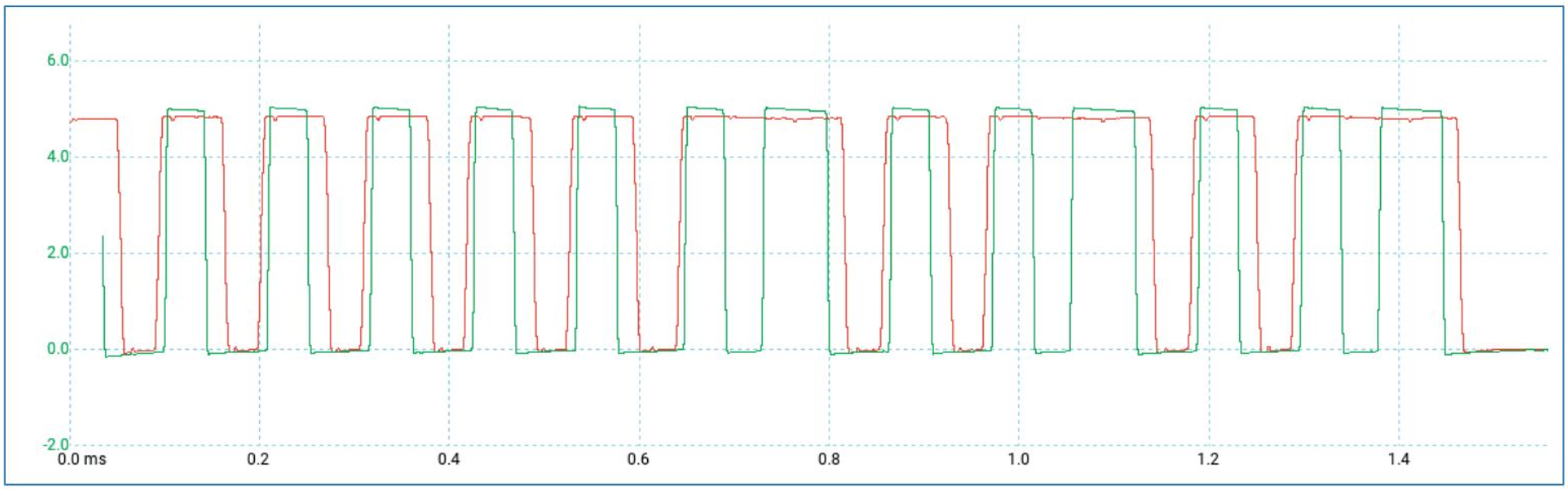
This could be partially due to the higher output power of our Transmitter, but there may also be some difference in the way that the incoming signal is modulated. Our findings in the Analog behaviour section above are consistent with that.

The next test involved speeding up the waveform until we started to see missed and distorted pulses. Our Transmitter did not miss a pulse until we reached cycle times under 95μs or around 10.5kbit/s, while the prebuilt module we purchased started missing pulses at around 105μs or 9.5kbit/s. Scope 5 shows the conditions we looked for.

The transmitter input is delayed to align it better to the receiver's output; you can see a few locations where the receiver waveform has stayed high when it should have gone low.

## SDR analysis

We also examined the output of the transmitters using a software defined



**Scope 5: the green trace is the transmitter input delayed slightly to roughly align it with the red trace of the receiver output. You can see that at higher pulse rates than in Scope 4, pulses are missed and the asymmetry between rising and falling edges is more pronounced.**

radio (SDR) receiver. The necessary hardware can be found in cheap USB TV receiver dongles. The free AirSpy SDR# program provides a way of receiving and viewing the spectrum of the RF signal.

We used a low-cost DVB-T dongle with an RTL2832U chip as the receiver. Since we were performing simple comparative tests, we used the basic whip antenna included with the dongle.

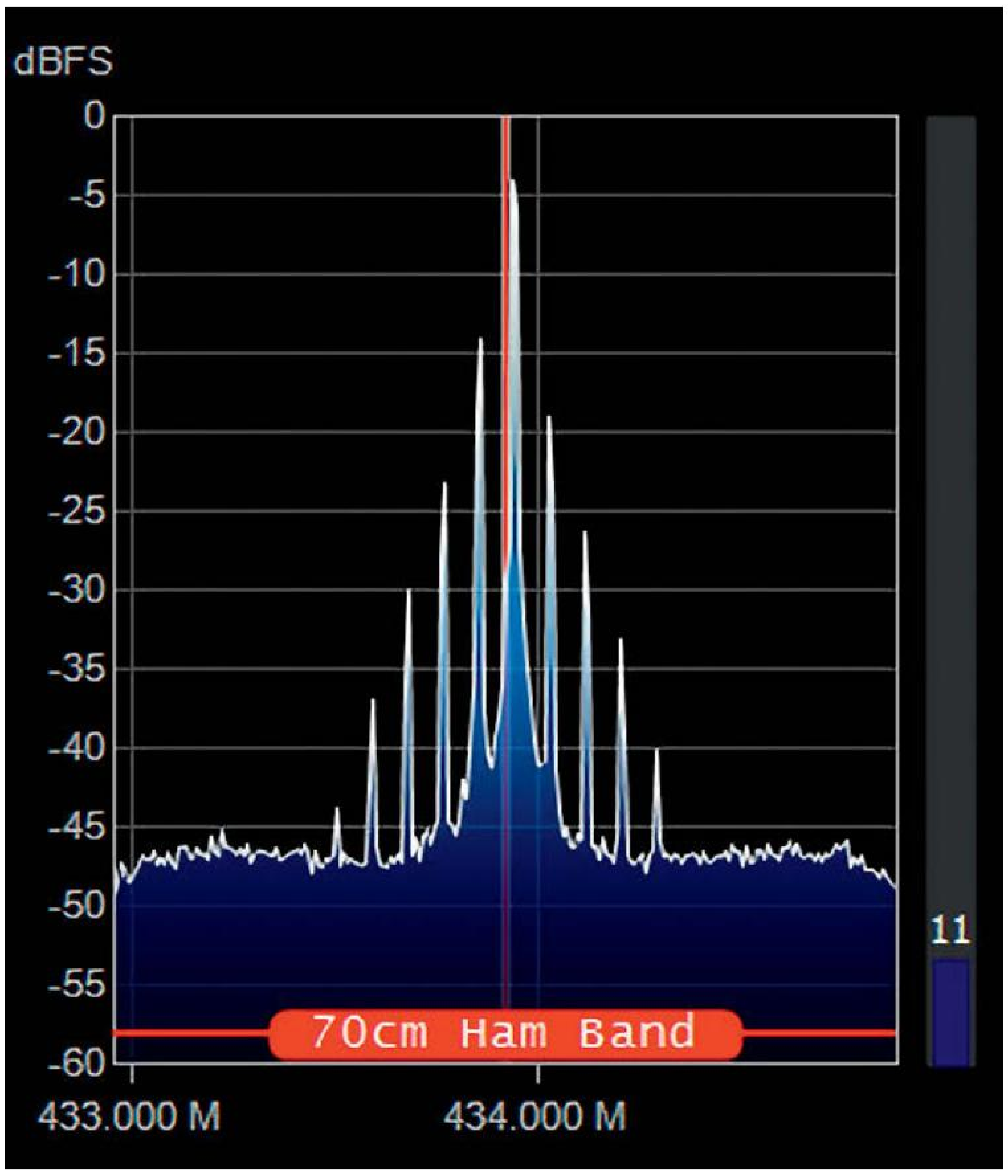
Screen 1 (our Transmitter) and Screen 2 (the one we bought) show the received spectra, with a displayed bandwidth covering the approximate range of the 433MHz LIPD band. The SDR is not a precision device, so the readings are simply relative to its full scale (0dB at the top of the spectra). The peak of both transmitters sits very close to 433.92MHz, as expected.

Our Transmitter has a few spurs and it peaks at around -4dB, while the commercial transmitter has a wider spread and more spurs. It peaks at around -8dB. That is not surprising, given the expected output power given by the respective data sheets.

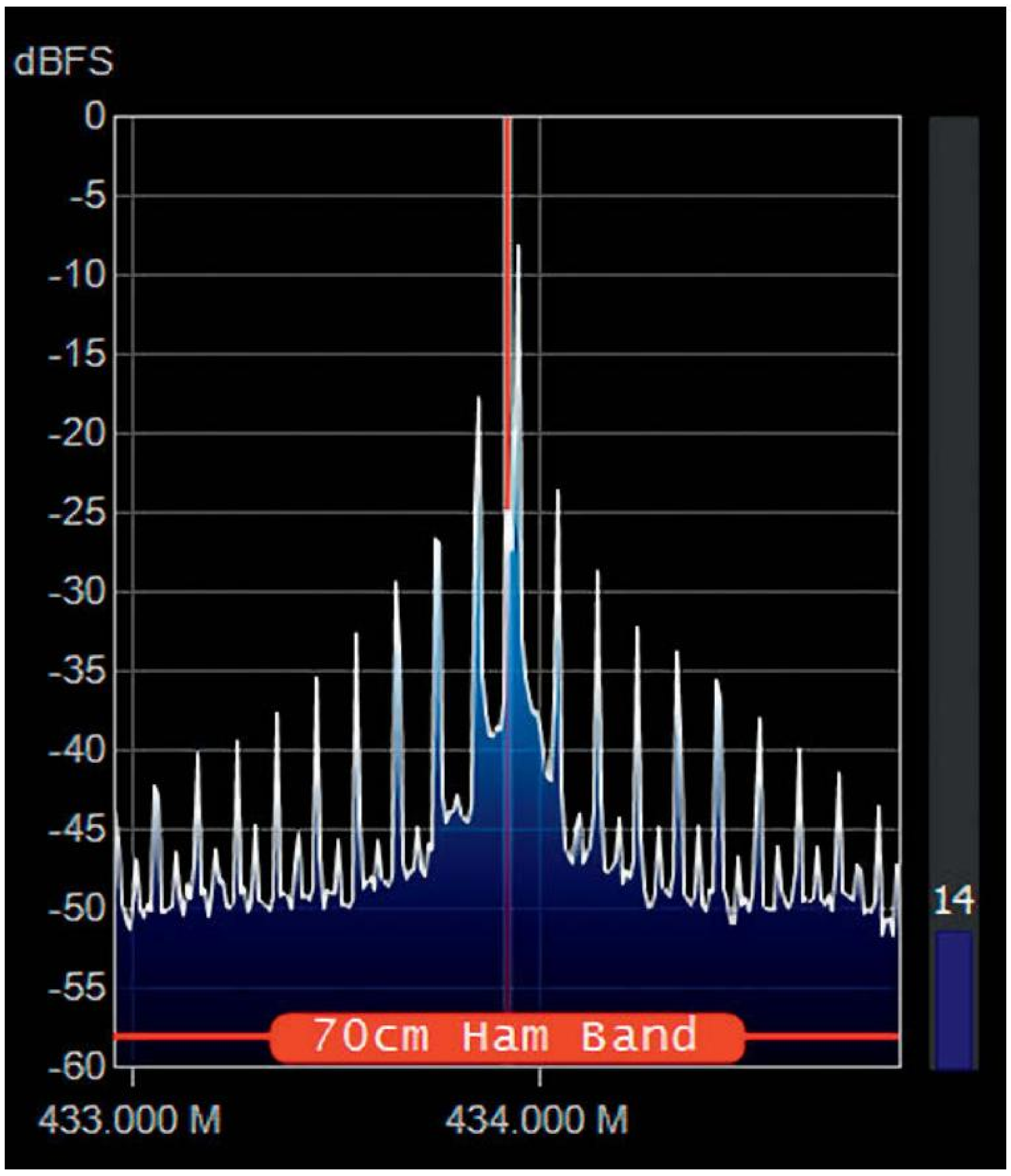
## Summary

Our Transmitter works as a drop-in replacement for a typical prebuilt transmitter; it should work in all applications that require a digital transmitter on the 433MHz LIPD band.

The only thing our module can't do that the prebuilt one can is transmit analog signals (ie, amplitude modulation). It works especially well for digital transmissions, though. PE



**Screen 1: the spectrum of our Transmitter is neatly confined to the 433MHz LIPD band.**



**Screen 2: the commercial transmitter module has a wider spread, more spurs and lower peak power than our Transmitter.**

# AOShop

**The home for specialist audio, analogue and historic components – provided by Jake Rothman, *PE's Audio Out* columnist. The AOShop is your best bet for classic analogue 'NOS' (new old stock) components, including all parts for *Audio Out* projects and designs.**

## Ge semiconductors

| | |
|---|---|
| *Small-signal PNP transistors* | |
| NKT214F, OC57, OC59, 2N1377, 2N525 | £1.00 |
| *Low-noise PNP transistors* | |
| GET106 | £2.50 |
| *Small power PNP transistors* | |
| AC153, AC153K, AC188 | £1.50 |
| *Small power NPN transistors* | |
| AC176, AC141K | £2.00 |
| *High-voltage PNP transistors* | |
| OC77, CV7001 | £1.50 |
| *TO3 PNP power transistors* | |
| OC22, CV7054 (OC23), OC25, OC35, OC36, AD143, AD149, AD161, AD162 | £2.00 |
| AD140 | £3.50 |
| AD149, AD161/2 matched pair | £5.00 |
| *RF PNP transistors* | |
| OC41, OC42, 2SA12, 2SA53,AF124, AF178, GET872A | £1.50 |
| *NPN transistors* | |
| OC139, OC140, ASY73 | £2.00 |
| *Diodes* | |
| CV7049 (OA10), CG92 (OA91) | £0.50 |

## Si semiconductors

| | |
|---|---|
| *Diodes* | |
| ZC5800 RF Schottky | £0.20 |
| *Low-noise silicon transistors* | |
| BFW16A, 2SC3071, 2SC3068, 2SA1016K, 2SC2362K, 2SA970BL | £1.50 |
| 2SC2204, 2SD655, BC550C | £0.50 |
| ZTX651 | £0.30 |
| ZTX751 | £0.50 |
| *RF transistors (suitable for Theremin)* | |
| BF199 | £0.50 |
| *Audio power MOSFETs* | |
| Exicon 10N20, 10P20 | £6.50 |
| Hitachi 2SJ99, 2SK343 | £3.50 |
| Hitachi 2SJ56, 2SK176 | £8.50 |
| *JFETs* | |
| BFW11, BFW10, TIS73L, J177, J113, U1994, U1898, 2SJ176, J201 | £1.00 |
| J175, J176, J112, J111, 2N3820, 2N5457, BF244, 2N5460, J230 | £0.60 |
| *Small power output/driver transistors* | |
| 2SB649A, D669A, 2SA1208, 2SC2910, MJE253G, MJE243G, 2SA1725, 2SC4511 | £1.20 |
| BD139,BD140, BD135, BD136, BD435, BD436, MPSA63, BCV46 | £0.50 |
| *MOSFETs* | |
| ZVP2106A | £0.30 |
| Dual-gate 3SK45, BFS28 | £1.50 |
| *Metal-cased transistors* | |
| BCY71 | £0.30 |
| *Amplifiers* | |
| LM318 high-speed op amp | £0.35 |
| µA709 metal-cased op amp | £2.00 |
| 5534H metal-cased op amp | £3.00 |
| LM384, TDA2030A, TDA2050V | £1.50 |
| TAA435 (Mullard power amp driver for AD161/AD162) | £1.20 |

## Synthesiser ICs

| | |
|---|---|
| That/dbx 2180 VCA/VCF | £6.00 |
| CA3080 VCA/VCF (vocoder) | £2.80 |
| CA3280 VCA/VCF | £4.00 |
| LM13600/ LM13700 VCA/VCF | £2.00 |
| CA3086 transistor array | £1.00 |
| PT2399 echo/delay | £1.50 |

## Dual transistors

| | |
|---|---|
| 2N2639, 2N2223, 2N2910 (NPN) | £4.00 |
| E401 (JFET Moog) | £4.00 |
| 2SK2145-Y dual JFET | £0.80 |
| 2N5564 JFET | £8.00 |
| HN3C51F, HN3A51F | £1.00 |
| DMMT3904/6, HN1A01F, HN1C01F | £0.50 |

## Loudspeakers

| | |
|---|---|
| *PE Mini-Monitor* | |
| Volt PE165 6.5-inch woofer (each) | £85 |
| Morel MDT29 tweeter (each) | £25 |
| Kit *pair* of PE165/MDT29 plus Volt crossover parts and PCBs | £299 |
| Monacor DT-28N tweeter (each) | £35 |
| Vifa 19mm BC20SC15-04 tweeter (each) | £15 |
| *Volt crossover inductors* | |
| 1.2mH, 1.5mH, 2mH, 2.7mH, 0.5mH (tapped at 0.3mH) (each) | £5.85 |
| *Fully assembled and tested high-quality speaker prototypes – ask for details* | |
| LS3/5As and other similar speaker systems (pair) | £200-£350 |
| *Fully tested reclaimed speakers* | |
| Vifa BC14 5-inch woofer (each) | £10 |
| Vifa TC26 1-inch tweeter (each) | £10 |
| *Low-price speakers* | |
| Philips 4-inch 4070 | £2.00 |
| EMI 10x6-inch, 30Ω Alnico | £7 |
| 64mm 64Ω neodymium | £1.20 |

## Capacitors

***Note '10/63' denotes '10µF 63V'.***

| | |
|---|---|
| *Polyester* | |
| 3.3/100, 4.7/250, 4.7/63 | £1.00 |
| 5.6/63, 8.2/63, 10/63 | £2.00 |
| Mullard 'Mustard' C296 0.22/400 | £2.00 |
| *Polycarbonate* | |
| Axial 2.2/63 1%, 4.7/160, 6.8/63 | £1.00 |
| *Radial* | |
| 6.8/160V, 10/63 | £2.00 |
| Reclaimed 22/63 | £2.00 |
| *Polystyrene* | |
| Philips 1% 4.7nF/160, 6.2nF/500, 12nF/63, 22nF/63, 110nF/63, 24nF, 2nF | £1.00 |
| RIFA 1% 100nF/100, Suflex 90.9nF 0.5% | £2.00 |
| Suflex 2.5% 10nF/63 (rad. or ax.) | £0.50 |
| *Radiation resistant* | |
| Siemens cellulose acetate MKL 2.2/25 | £0.80 |
| *Electrolytic – Mullard blue 017 series* | |
| 10/25, 22/25, 100/10 | £0.50 |
| 150/40, 470/40, 1000/40 | £1.00 |
| *Tantalum – axial metal cased* | |
| 22/50, 47/35, 68/25, 100/20, 120/10, 150/16, 220/10, 330/6 | £2.00 |
| 22/35, 33/35, 47/20, 68/15, 100/10, 150/6 | £1.25 |
| 4.7/50, 6.8/35, 10/25, 10/35, 22/15 | £1.00 |
| *Axial moulded-case tantalum* | |
| Kemet axial 6.8/10 | £0.30 |
| Kemet radial 33/10 | £0.40 |
| STC radial 100/20 | £1.50 |
| *Tantalum bead* | |
| 22/50 | £1.00 |
| 470/3 | £2.00 |
| 680/6.3 | £3.00 |
| *Wet tantalum* | |
| 220/25 axial | £3.20 |
| Castanet button 140/30, 470/3 | £3.20 |
| Hughes 540/10 | £3.20 |
| *Bipolar* | |
| Hermetic bipolar tantalum 16/35 | £3.20 |
| Elcap axial 10/50 | £0.50 |
| Generic radial 100/16, 470/35, 100/ 63, 22/35, 4.7/35, 220/16 | £0.50 |
| *Philips solid-aluminium (axial) 121/123* | |
| 47/16 | £1.00 |
| 330/6.3 | £2.00 |
| 100/35 | £4.00 |
| *Philips Pearl 122 series (radial)* | |
| 10/16 | £0.30 |
| *Silvered mica (radial)* | |
| 1nF/500 1% | £1.00 |
| *Trimmer capacitors* | |
| Vishay plastic-film 4-40pF | £0.80 |
| Vishay plastic-film 5.5-45pF | £1.00 |
| Vishay plastic-film 5-80pF | £1.20 |
| Vishay plastic-film 10-250pF | £2.00 |
| Mica 1-12pF, 2-40pF | £1.00 |

## Audio transformers and inductors

| | |
|---|---|
| *Eagle transformers* | |
| LT44, LT722 driver, LT700, LT723 500Ω output | £2.50 |
| LT30 500mW output | £3.50 |
| 5:1 interstage (fuzz box) | £1.50 |
| *Repanco* | |
| T/T3 splitter transformer | £4.00 |
| CH2 5mH RFC | £2.00 |
| *Balanced output transformer* | |
| Vigortronix 600Ω VTX-101-007 | £10 |
| Vigortronix 600Ω VTX-101-3001 | £10 |
| Vigortronix 600Ω VTX-101-3002 | £15 |
| Gardners 150Ω | £10 |
| Reclaimed BBC LL74/MPC nickel core 600Ω | £12 |
| **Low-level 1:1** 600Ω (modem) | £2 |
| Reclaimed mic input transformer | £15 |
| *Inductors* | |
| 82µH, 4.7mH, 100µH, 270µH, 10µH, 14µH (low Z) | £0.50 |
| 4-inch ferrite rod with MW and LW windings | £3.00 |

## Special resistors

| | |
|---|---|
| Bourns wire-wound trimmer 5kΩ 3059 JM panel-mount | £2.00 |
| Thermistor RA53, R13 | £4.00 |
| A13 | £2.00 |
| Thermistor CZ1, CZ6 | £1.50 |
| Holco H2 2.2MΩ 1W, 1% | £1.00 |
| Welwyn 1GΩ 2W | £1.00 |
| 5k Bourns 3321H cermet trimmer | £0.50 |

## Potentiometers

| | |
|---|---|
| *Bourns 81* | |
| 25kΩ lin cermet | £2.00 |
| 25kΩ lin, 5kΩ lin cond. plastic | £3.00 |
| *Bourns 91* | |
| 10k dual-gang lin or log | £10.00 |
| *Plessey moulded-track* | |
| 5kΩ log with switch | £3.00 |
| 50kΩ A/log | £3.00 |
| Mil 250kΩ lin dual | £5.00 |
| *Alpha* | |
| 16mm 4.7kΩ A/log | £0.80 |
| 220kΩ A/log | £0.80 |
| 10kΩ lin centre-detent dual-gang | £1.50 |
| *Allen Bradley J series/Honeywell* | |
| 10kΩ lin 1 million cycles | £5.00 |
| Blore Edwards AB 45 dual 5kΩ A/log with switch | £3.50 |
| Alps RK9 dual-gang 5kΩ RD law | £4.00 |
| Alps 50k log tapped motor. stereo | £5.00 |
| BI P260 500kΩ log conductive plastic 1 million cycles | £2.00 |
| Colvern wire-wound 100kΩ or 50kΩ dual-gang 3W | £5.00 |
| Mil Spec hermetic 10Ω | £8.00 |

## Miscellaneous

| | |
|---|---|
| *PCBs* | |
| All-pass Filter PCBs (three) | £4.00 |
| .. Rotary Switch (AO0626ROT) | £1.50 |
| .. Pot Board (AO0626POT) | £1.50 |
| Silent Stomp Switch (pair) | £4.00 |
| Penta-fuzz Bender (*PE*, Feb26) | £3.00 |
| Pre-driver (*PE*, Jan26) | £3.00 |
| Pre-driver jack bd (*PE*, Jan26) | £1.50 |
| Fuzz Box (*PE*, Nov25) | £3.00 |
| Fuzz Box jack bd (*PE*, Dec25) | £1.50 |
| Dual JFET Buffer (*PE,* Oct25) | £2.00 |
| Mosfet Buffer (*PE,* Oct 2025) | £2.00 |
| Dual Mosfet Buffer (*PE,* Oct25) | £2.00 |
| Cascode Amp (*PE*, Sep 2025) | £2.00 |
| Cascode PS (*PE*, Sep 2025) | £2.00 |
| Pocket Theremin (*EPE*, 1996) | £2.00 |
| Elysian Theremin (*EPE*, 1996) | £6.00 |
| Synth VCF, VCO (*EPE*, 2017) | £3.00 |
| 48V PSU (*EPE*, 2019) | £3.00 |
| SMT dual transistor adapter | £0.40 |
| *Relays* | |
| STC hermetic 4190 24V DPDT | £4.00 |
| *Switches Blore Edwards/Plessey* | |
| MH1 4-pole 3-way | £3.00 |
| Elma 01 2-pole 6-way | £6.00 |
| 3PDT guitar pedal stomp switch | £2.50 |
| *Valves* | |
| EC88, PCL84 | £3.00 |

**Contact**

Jake Rothman
Manchester House
Market Street, Craven Arms
Shropshire SY7 9NN
(visit by appointment)
+44 (0)1597 829102
jrothman1962@gmail.com

Minimum order £5.00 inc post

**Payment**

PayPal, bank transfer, cheques (payable to "J Rothman", UK pounds only). No VAT payable

**Postage**

Small Jiffy bag £2.99
Small package £4.99
Big boxes and overseas at cost – ask for a quote

# AUDIO OUT

By Jake Rothman

## All-pass filters, part two – time delays from audio filters

Along with the all-pass filters described last month, ordinary low-pass and high-pass filters also generate a group delay, as shown in Fig.20. This can either be a useful or an undesirable feature. A very rough approximation for filter delay can be achieved by taking the reciprocal of half the maximum phase shift (180°) at the cut-off frequency.

So for a 1kHz fourth-order filter, the group delay would be 1 ÷ $2f$ = 500µs. Lower-order filters will have less delay. Bessel filters have a flat group delay, whereas the more common Butterworth type (which has a flat passband response) has a small hump in delay at the roll-off point.

If the outputs of a high-pass filter and a low-pass filter are combined, the delay remains, resulting in an all-pass filter, as shown in Fig.21. In loudspeaker crossover networks, the same effect occurs, although the summation is acoustic.

In another example, the second-order state-variable filter is a common building block of many audio systems, providing low-pass, high-pass and band-pass outputs. By adding the low-pass and high-pass outputs together, a notch filter is formed because the outputs cancel where the frequencies overlap. If these outputs are subtracted using a differential amplifier, an all-pass output results.

The whole system is shown in Fig.22 and Photo 11. A cheap way of obtaining a variable all-pass filter in the studio is to combine the outputs of an off-the-shelf active crossover.

These units are normally based on fourth-order state-variable filters with Linkwitz-Riley curves (-6dB at the crossover frequency); some typical rack-mountable adjustable crossover units are shown in Photo 12.

### All-pass filters in equalisers

The state-variable filter, although incredibly useful, uses integrators. These can suffer instability at DC and high frequencies with some op amps. I've had this a few times with my favourite, the NE5532, in parametric equalisers.

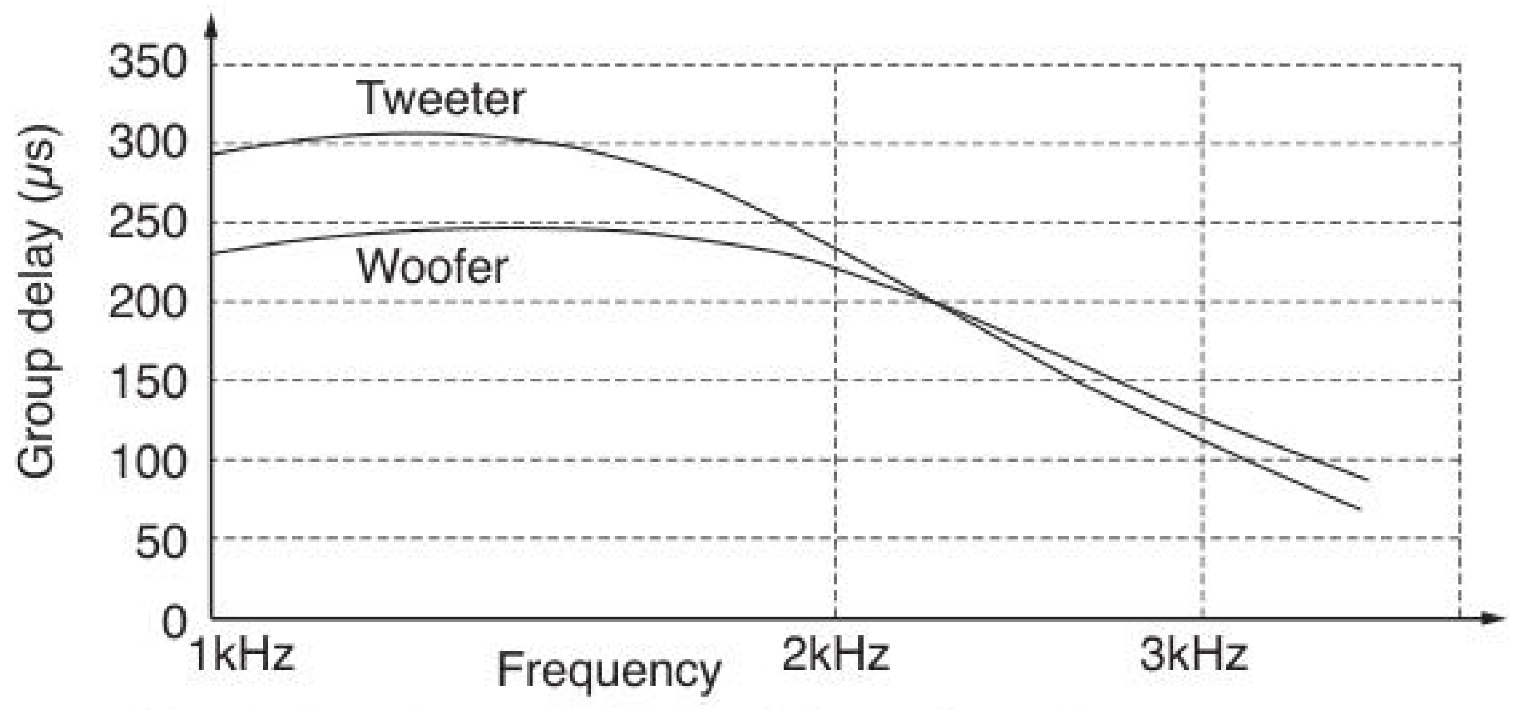


Fig.20: the group delay for the 1.8kHz fourth-order Linkwitz Riley high-pass and low-pass filters used in the TI active loudspeaker.

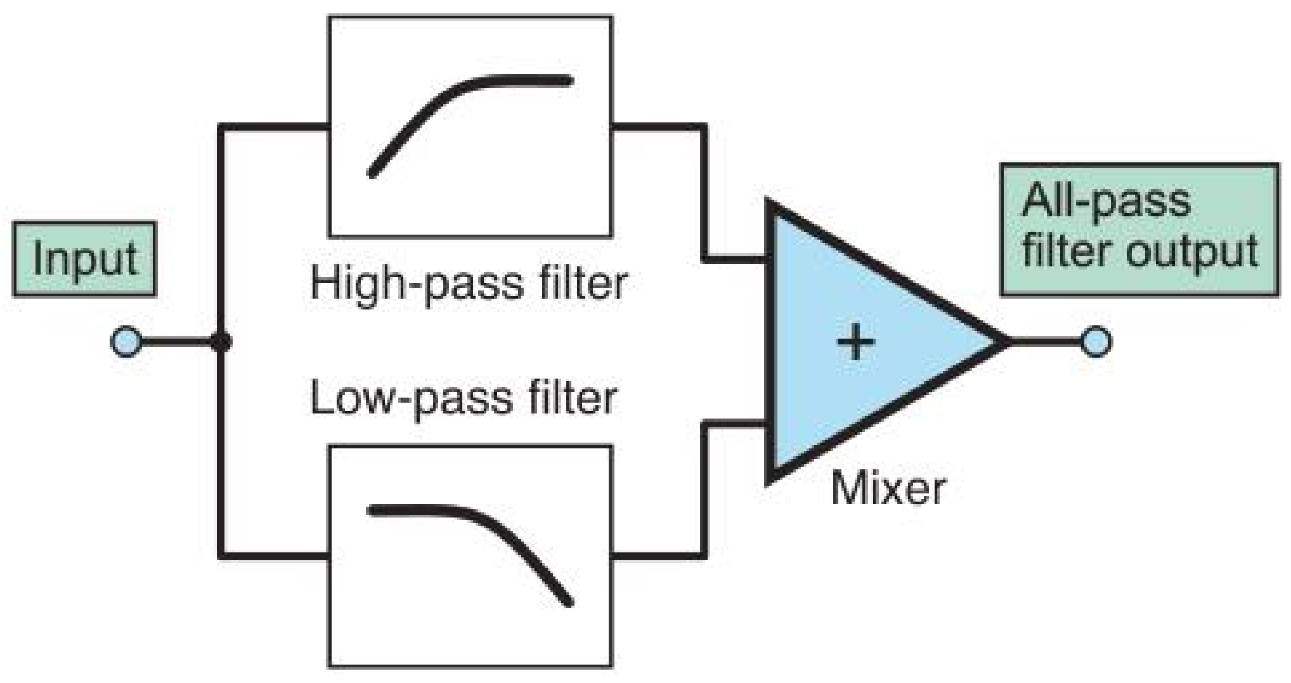

Input
High-pass filter
Low-pass filter
+
Mixer
All-pass filter output

Fig.21: mixing the outputs of a high-pass and low-pass filter gives an all-pass filter.

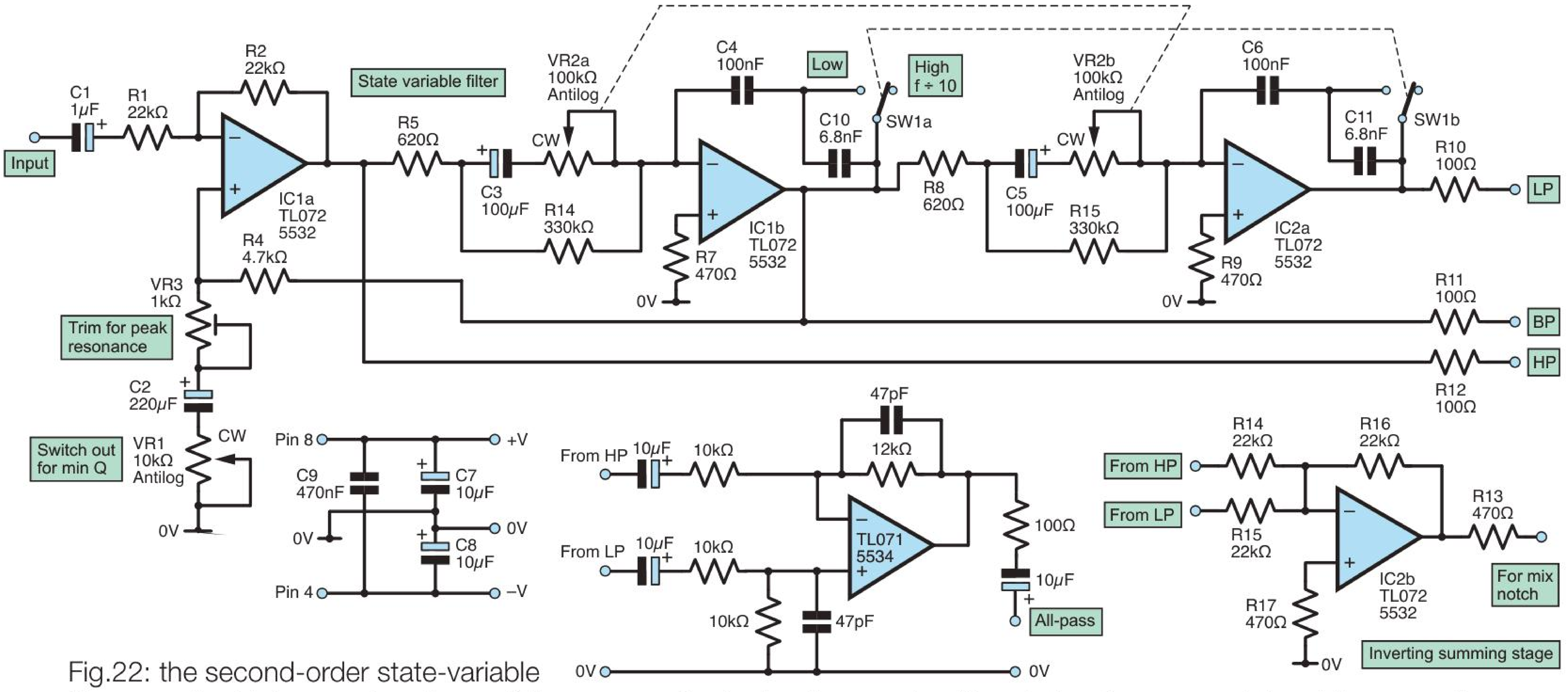

Input
C1 1µF
R1 22kΩ
R2 22kΩ
IC1a TL072 5532
State variable filter
R4 4.7kΩ
VR3 1kΩ
Trim for peak resonance
C2 220µF
VR1 10kΩ Antilog
CW
Switch out for min Q
0V
R5 620Ω
C3 100µF
VR2a 100kΩ Antilog
CW
R14 330kΩ
C4 100nF
Low
C10 6.8nF
High f ÷ 10
SW1a
IC1b TL072 5532
R7 470Ω
0V
R8 620Ω
C5 100µF
VR2b 100kΩ Antilog
CW
R15 330kΩ
C6 100nF
C11 6.8nF
SW1b
IC2a TL072 5532
R9 470Ω
0V
R10 100Ω
LP
R11 100Ω
BP
R12 100Ω
HP
Pin 8
+V
C9 470nF
C7 10µF
0V
C8 10µF
0V
Pin 4
–V
From HP
10µF
10kΩ
47pF
12kΩ
From LP
10µF
10kΩ
TL071 5534
10kΩ
47pF
100Ω
10µF
All-pass
0V
0V
From HP
From LP
R14 22kΩ
R16 22kΩ
R15 22kΩ
R13 470Ω
IC2b TL072 5532
R17 470Ω
0V
For mix notch
Inverting summing stage

Fig.22: the second-order state-variable filter can give high-pass, band-pass & low-pass outputs simultaneously with only two frequency determining capacitors. Notch and all-pass outputs can be obtained by summation/subtraction of the high-pass and low-pass outputs.

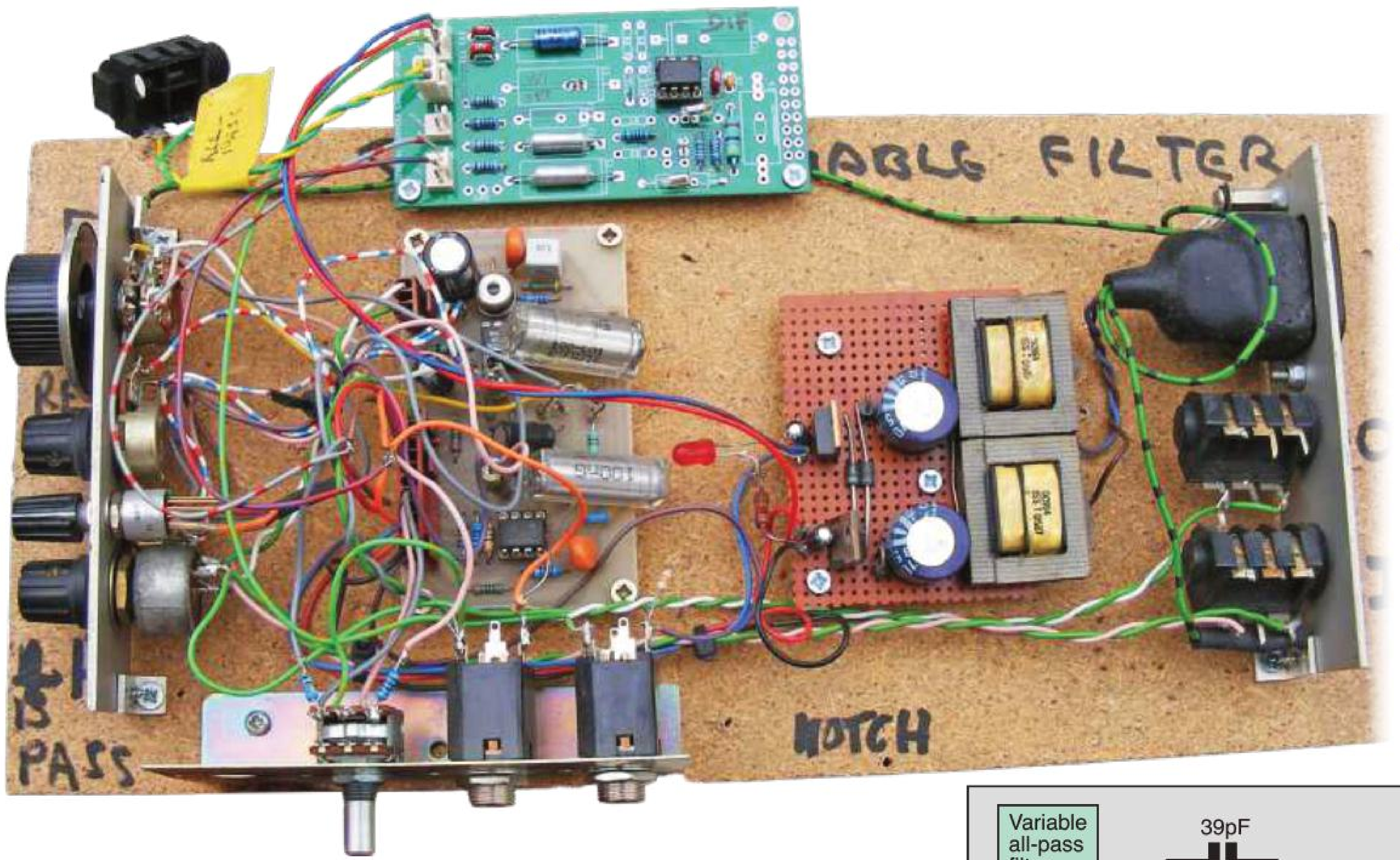

ABLE FILTER
NOTCH
PASS

Photo 11: a lecture lab demonstration set-up of the state variable filter shown in Fig.22.

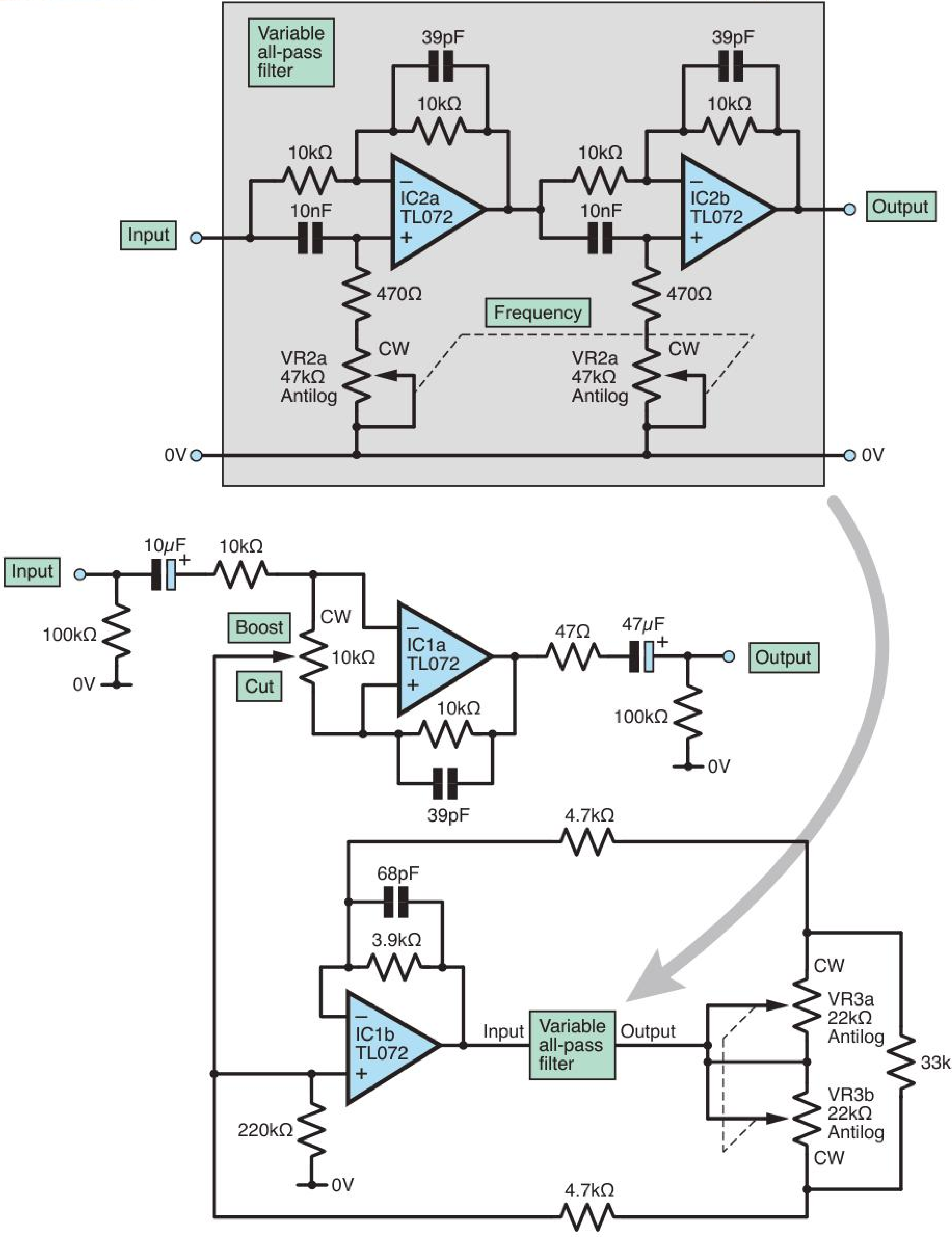
Steve Dove, one of the most influential mixing console designers, replaced the state-variable filter in his EQ with a pair of all-pass filters, using the arrangement shown in Fig.23. This allowed each op amp to be individually stabilised, rather than via an overall loop; 10kΩ feedback resistors and the 39pF phase-lead capacitors do the job.

AC-coupling capacitors (with DC biasing resistors) can then be put in series with the pots to prevent scratching without the risk of latch-up.

This circuit configuration never really took off because of the dual-gang anti-log Bourns 91A potentiometers specified, which were very expensive. I think it is worth revisiting now that equivalent cheap pots are readily available, making a few tweaks and creating a PCB.

Of course, such a circuit with lots of positive feedback becomes an oscillator. A similar all-pass system was used in Marconi audio analysers.

Fig.23: a musical parametric equaliser using all-pass filters designed by Steve Dove. It is more stable than the more common SVF type using dual integrators.


Variable all-pass filter
39pF
10kΩ
10kΩ
10nF
IC2a TL072
–
+
39pF
10kΩ
10kΩ
10nF
IC2b TL072
Input
Output
470Ω
470Ω
Frequency
VR2a 47kΩ Antilog
CW
VR2b 47kΩ Antilog
CW
0V
0V
Input
10μF
10kΩ
100kΩ
0V
Boost
Cut
CW
10kΩ
IC1a TL072
10kΩ
39pF
47Ω
47μF
Output
100kΩ
0V
4.7kΩ
68pF
3.9kΩ
IC1b TL072
Input
Variable all-pass filter
Output
CW
VR3a 22kΩ Antilog
VR3b 22kΩ Antilog
CW
33kΩ
220kΩ
0V
4.7kΩ

## Loudspeaker phase problems

Multi-way loudspeakers are the greatest embodiment of phase and polarity problems in audio.

PULSE
CX-PRO
STEREO CROSSOVER
CHANNEL 1
CHANNEL 2
SUPER-X PRO
BEHRINGER

Photo 12: a selection of low-cost Chinese active crossover units. At the top is a third-order unit from Pulse; the lower black units are fourth-order LR crossovers from Behringer.

Fig.24 shows the seven phase variables in a typical two-way speaker. A three-way system is even worse. The anti-phase nulls around the crossover region can be clearly heard with white/pink noise if one bobs one's head up and down in front of the speaker.

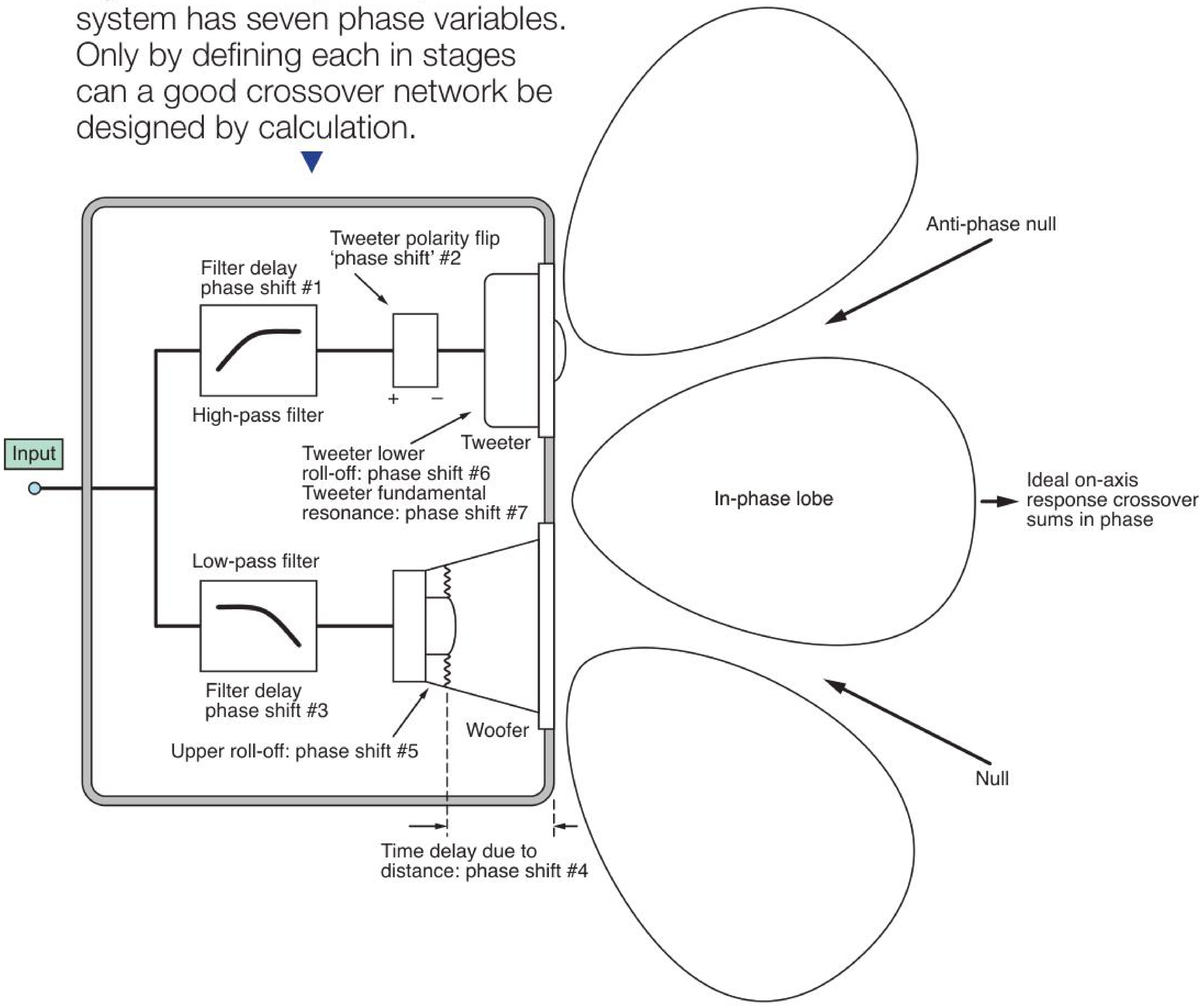

Input
Filter delay phase shift #1
Tweeter polarity flip 'phase shift' #2
+ −
High-pass filter
Tweeter
Tweeter lower roll-off: phase shift #6
Tweeter fundamental resonance: phase shift #7
Low-pass filter
Filter delay phase shift #3
Upper roll-off: phase shift #5
Woofer
Time delay due to distance: phase shift #4
Anti-phase null
In-phase lobe
Ideal on-axis response crossover sums in phase
Null

Fig.24: a two-way loudspeaker system has seven phase variables. Only by defining each in stages can a good crossover network be designed by calculation.

If a square wave is put through a multi-way loudspeaker, it invariably emerges as a scrambled mess due to its complex overall phase response. In the 1970s, there was a craze of attempting to make "linear phase" loudspeakers. While never achieved, this did provide a significant impetus to crossover development.

These speakers had a strange appearance, having stepped fronts in an attempt to provide phase alignment on the listening axis. The B&W 801 and KEF 105 were denigrated as looking like the Daleks from Dr Who, but their crossovers were superbly smooth. Later on, these physical manifestations of phase correction disappeared as crossover designers added all-pass filters.

The loudspeaker system I described in part one (Fig.16 last month) is shown in Photo 13 and looks completely conventional. Note the crossover with the hard-wired all-pass filter (with yellow capacitors and white inductors) on top.

Photo 13: this little LS3/5a derivative, using a Morel MDT29 tweeter instead of a KEF T27, has an all-pass filter to allow a conventional baffle with flush front-mounted drivers to be used.

However, all-pass filters are still rare in passive crossover systems. One I am familiar with was the Tannoy Broadcast Monitor 8, made in the 1980s. This was the monitoring system we had at the university studio when I was an undergraduate.

Of course, I had to take it to bits to trace the crossover circuit, with the result shown in Fig.25. The larger delay needed for matching a 25mm dome tweeter to a 200mm bass/mid unit was achieved with a second-order all-pass network with four inductors. I've noticed that time-aligned systems such as this have superior stereo imaging.

## Drive unit polarity

It's essential to connect the woofer correctly, with its positive terminal being driven from the positive connector (on the back of the box) to ensure consistent absolute polarity. The mid-range and tweeters are often

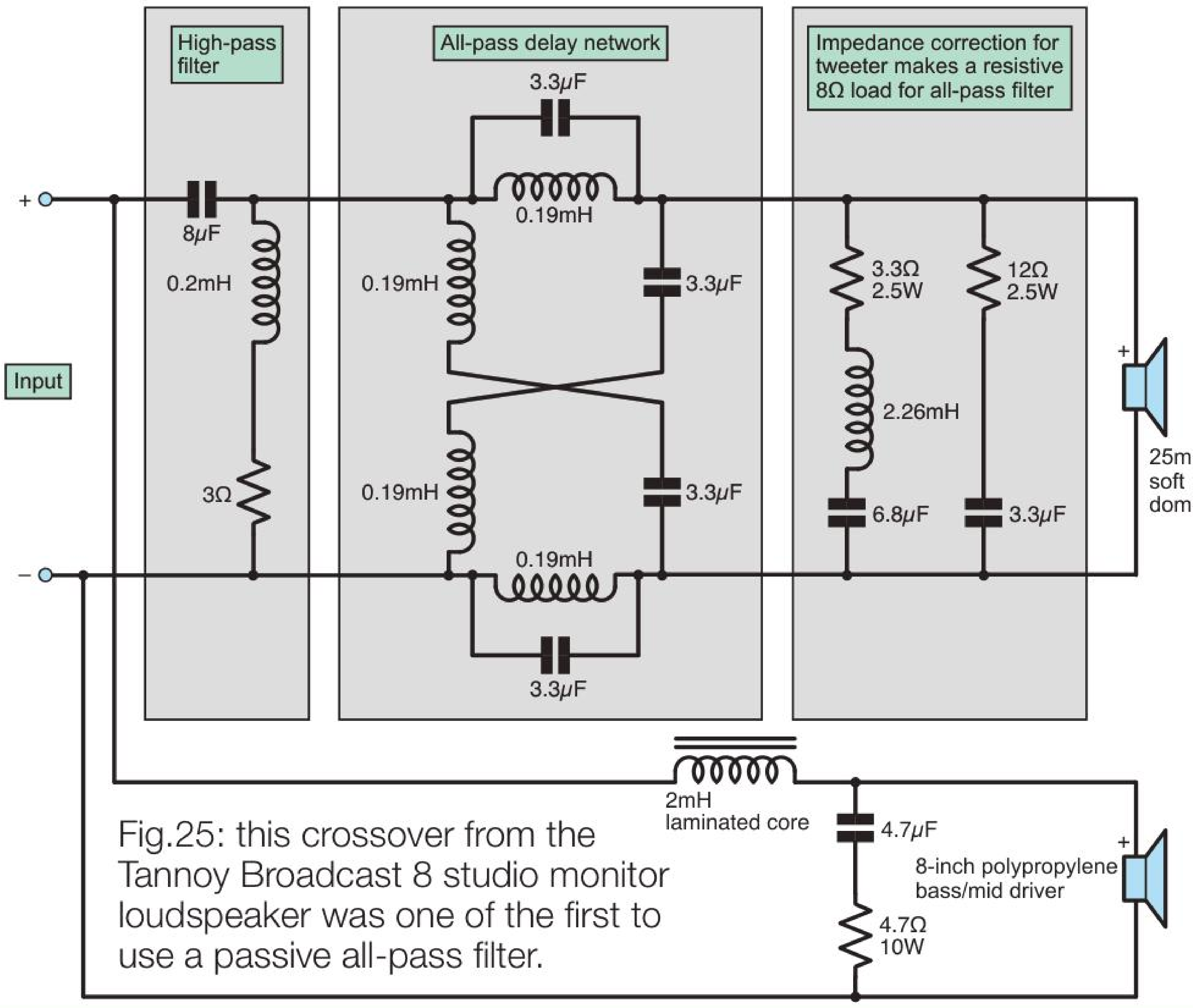

High-pass filter
All-pass delay network
Impedance correction for tweeter makes a resistive 8Ω load for all-pass filter
+
−
Input
8μF
0.2mH
3Ω
3.3μF
0.19mH
0.19mH
3.3μF
0.19mH
3.3μF
0.19mH
3.3μF
3.3Ω 2.5W
12Ω 2.5W
2.26mH
6.8μF
3.3μF
+
25mm soft dome
2mH laminated core
4.7μF
+
8-inch polypropylene bass/mid driver
4.7Ω 10W

Fig.25: this crossover from the Tannoy Broadcast 8 studio monitor loudspeaker was one of the first to use a passive all-pass filter.

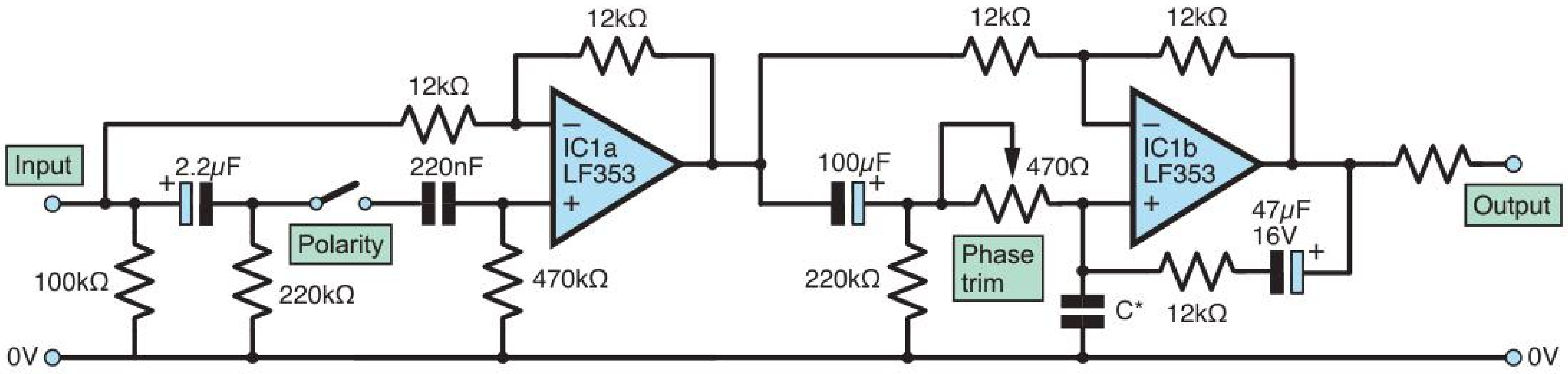


Fig.26: the Brooke Siren Systems FDS 360 was one of the first active crossovers to use all-pass filters.

| *f* | *C* |
|---|---|
| 100 – 200 | 68nF |
| 220 – 400 | 47nF |
| 500 – 800 | 22nF |
| 1kHz – 1.3kHz | 15nF |
| 1.5kHz – 2kHz | 10nF |
| 2.5kHz – 4kHz | 4.7nF |
| 4.5kHz – 7kHz | 3.3nF |

*Select *C* according to crossover frequency *f*

wired in reverse polarity to 'correct' filter phase shift and physical offset problems.

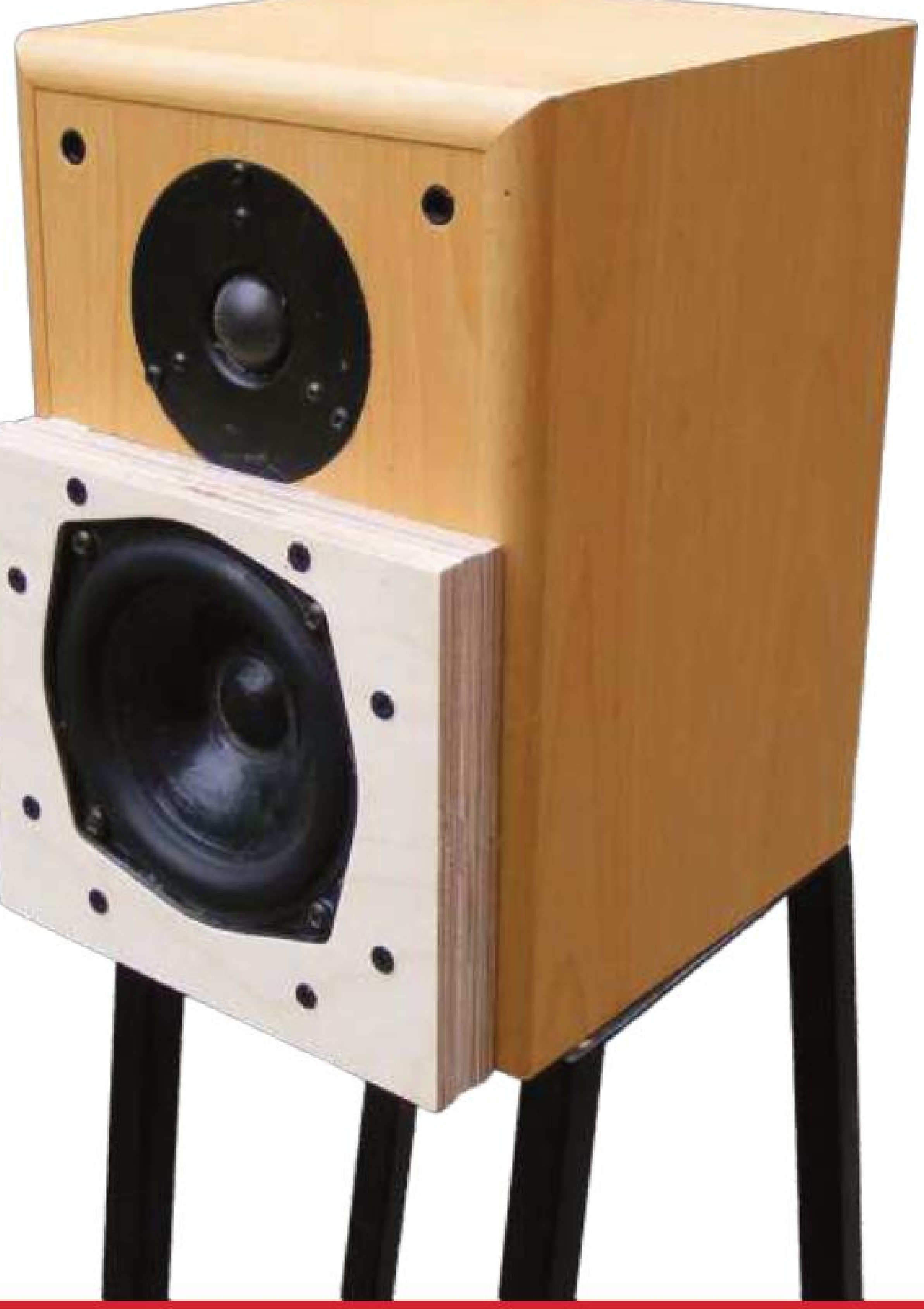
This is a naughty bodge, since we are using a polarity inversion to compensate for a phase shift, which is not mathematically correct. You can't theoretically correct a time delay problem with a polarity inversion. However, it can eliminate a notch reasonably well over a small frequency band, such as the crossover overlap.

The tweeter is often inverted with 2nd-order crossovers, where the low-pass and high-pass outputs are theoretically 180° apart in phase. With fourth order, they are 360° apart, which gives the illusion they are of the correct 'polarity' again. One has to be careful though, with the off-axis dips that may occur just outside the crossover area.

A similar scenario exists where all-pass filters are used to compensate for driver displacement and filter phase shift, since they also operate over a finite frequency range.

Going back to my bad habit of taking everybody's stuff apart. I did it with my then-girlfriend's, now wife's, KEF Coda speakers back in 1993. Unfortunately, I put one back together with the tweeter wires flipped and she noticed. I was secretly miffed that her hearing was better than mine; I now get her to listen to everything I make.

Photo 14: a stepped baffle can be used to align the acoustic radiation planes of the woofer and tweeter, but it's a bit of a faff to build and ugly. ▼



## Speaker phase response

When the acoustic outputs of speaker crossovers filters are combined with the drive units' responses, the all-pass response is almost chaotic, usually with a large additional increase in group delay around the crossover point.

At higher frequencies, around the typical mid tweeter crossover range of 1.8kHz to 4kHz, the sonic effect of all-pass filtering has been shown to be practically inaudible by Jim Wilkinson (Bookshelf Loudspeaker Mk II, *Wireless World*, June 1979) and Siegfried Linkwitz (**www.linkwitzlab.com**).

However, at low frequencies (<700Hz), it becomes noticeable due to the longer time delay for the same number of cycles shift.

## Stepping back

The old way to obtain time alignment with the bass/mid unit is to position the tweeter behind the woofer to line up the drivers' acoustic centres, as shown in Photo 14, by using a stepped baffle. This was first done for the UK home constructor in Jim Wilkinson's *High Quality Bookshelf Loudspeaker* (*Wireless World*, October 1977).

I remember struggling with the woodwork for this design aged 15 at school in Suva, Fiji, in 1978. The teacher thought I was insane, saying in his Indian accent, "Why you build silly box?" I didn't bother to explain. Such steps are generally not a good idea because the reflections and diffraction anomalies wipe out the gains. Applying thick felt helps, but it looks bad. 48 years later, the WW speakers still live on in a studio in Bath (I'm 63 now).

It's better engineering to use an all-pass filter to delay the signal to the tweeter. A more recent design also worth reading is the *Analog Active Crossover Circuit for Two-Way Loudspeakers* by John Caldwell for Texas Instruments (TI).

## All-pass filter time correction

Several stages of all-pass filtering, typically two or three, are needed to get the required delay for most tweeters. This is because there is often a droop in group delay with all-pass filters at the higher frequency end. The number of stages can often be reduced if a tweeter polarity flip is introduced and the group delay of the filters is added in.

The general approach in the industry for all-pass filter stages seems to be to make the turnover frequency of the all-pass filters ($1 \div 2\pi RC$) around three to four times the crossover frequency. This means more stages are needed because of the higher frequency.

If you actually knew the group delay of the filters, the group delay of the drivers (due to their roll-offs) as well as the physical displacement, the calculations could be performed.

KEF managed all this data crunching with a Hewlett-Packard mainframe computer in the late 1970s. They included all the parameters necessary to generate an electrical filter that would give the desired electroacoustic target function, such as a Butterworth curve. This was the combination of the electrical and acoustic response, giving a predictable crossover result.

The process is described in an article titled *Crossover Networks* in the December 1981 issue of *Electronics Today International* (ETI). It's still very apparent from most speaker designs today that this lesson has "still not been learned", as a UK civil servant would say, after 45 years.

Many designers still think the crossover electrical slope defines the speaker slope, hence all the low-cost crossover units just seem to have Linkwitz-Riley Butterworth slopes. They only work if all your drivers have a DC-to-light frequency response.

One exception was the Brooke Siren Systems FDS 360 crossover, where you could have any slope or frequency selected by a plug-in card. It was also the only one at the time with variable all-pass filters on every output, as shown in Fig.26. It could minimise nulls that the polarity switch could not deal with. Later

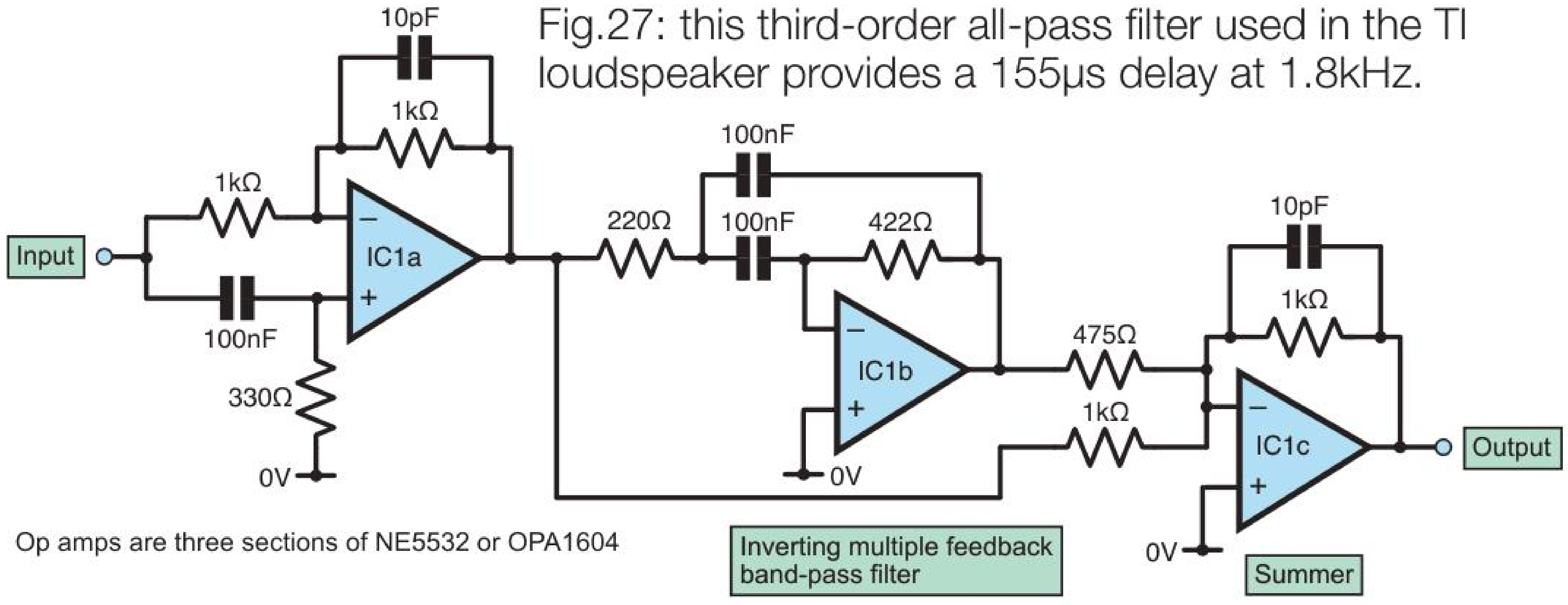

10pF
1kΩ
1kΩ
Input
100nF
330Ω
0V
IC1a
220Ω
100nF
100nF
422Ω
IC1b
0V
475Ω
1kΩ
10pF
1kΩ
IC1c
0V
Output
Op amps are three sections of NE5532 or OPA1604
Inverting multiple feedback band-pass filter
Summer

Fig.27: this third-order all-pass filter used in the TI loudspeaker provides a 155µs delay at 1.8kHz.

they made a digital unit that could provide any delay required.

One all-pass stage wasn't quite enough for some systems, but it was a useful trim after the driver boxes had been sensibly positioned. The FDS 360 crossovers were used for the Turbosound PA system. They had a higher frequency and shallower slope on the mid-range horn low-pass filter compared to the tweeter high-pass. When I worked there in the late 1980s, I enjoyed setting up the 360s.

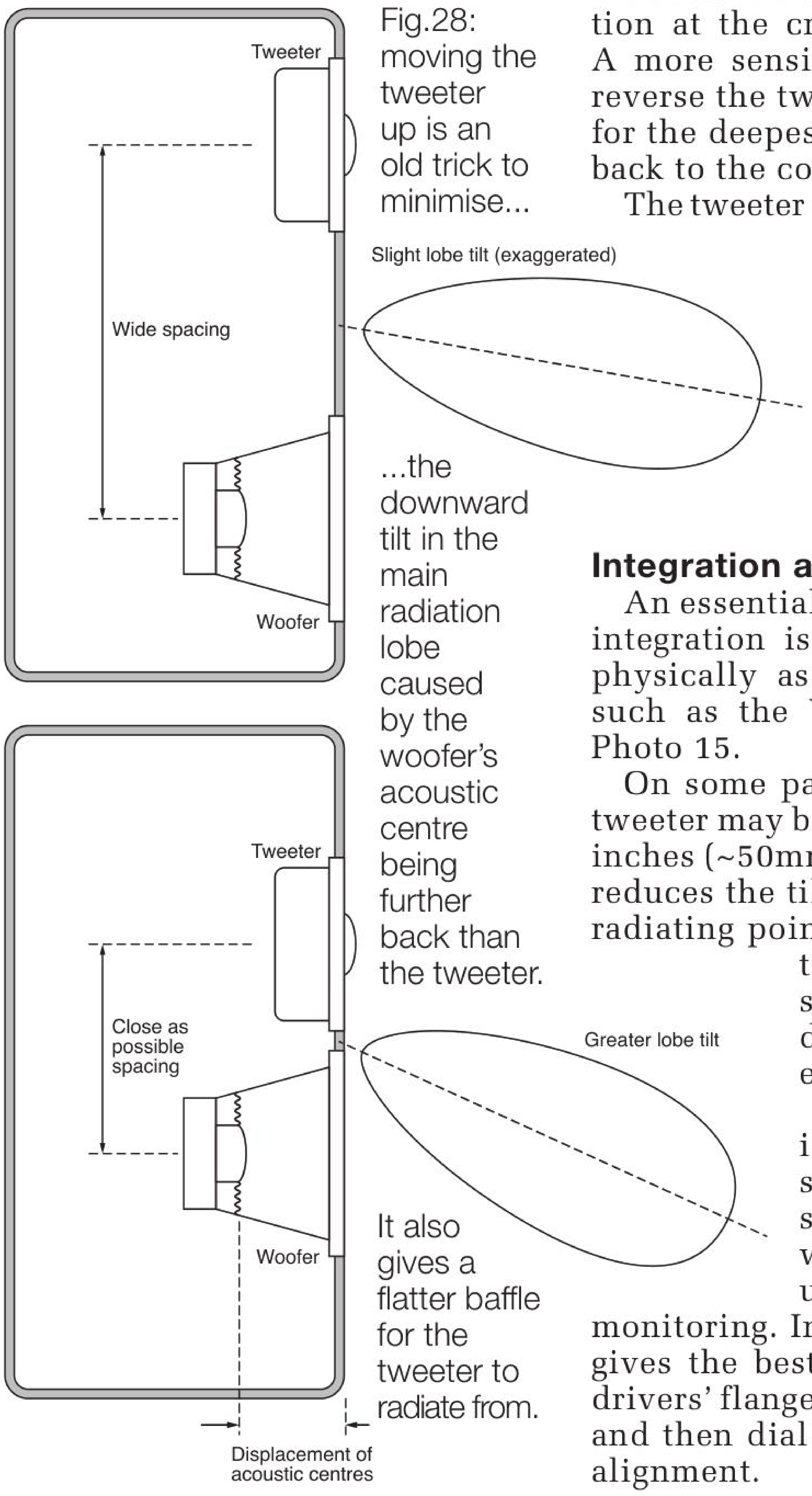

Tweeter
Wide spacing
Woofer
Slight lobe tilt (exaggerated)
Tweeter
Close as possible spacing
Woofer
Greater lobe tilt
Displacement of acoustic centres

Fig.28: moving the tweeter up is an old trick to minimise... ...the downward tilt in the main radiation lobe caused by the woofer's acoustic centre being further back than the tweeter. It also gives a flatter baffle for the tweeter to radiate from.

## Time delay compensation

To calculate the distance to be compensated for, the familiar time equals distance divided by velocity equation is used, or $T = D \div c$, where c is the speed of sound, which is about 343m/s at sea level with standard temperature and pressure. This allows a rough estimate to be made.

It's best to take the measurements from the drivers' actual acoustic centres, or by sliding the tweeter back and forth relative to the bass/mid unit for best summation at the crossover frequency. A more sensitive test here is to reverse the tweeter polarity, tune for the deepest notch, then put it back to the correct polarity.

The tweeter delay circuit used in the TI design combined second-order and first-order all-pass stages to give the required delay of 155µs at 1.8kHz, as shown in Fig.27.

## Integration and design

An essential part of good driver integration is to get the drivers physically as close as possible, such as the Wavecor design in Photo 15.

On some passive speakers, the tweeter may be placed a couple of inches (~50mm) higher because it reduces the tilt angle of the main radiating point. A quick application of trigonometry, shown in Fig.28, demonstrates this effect.

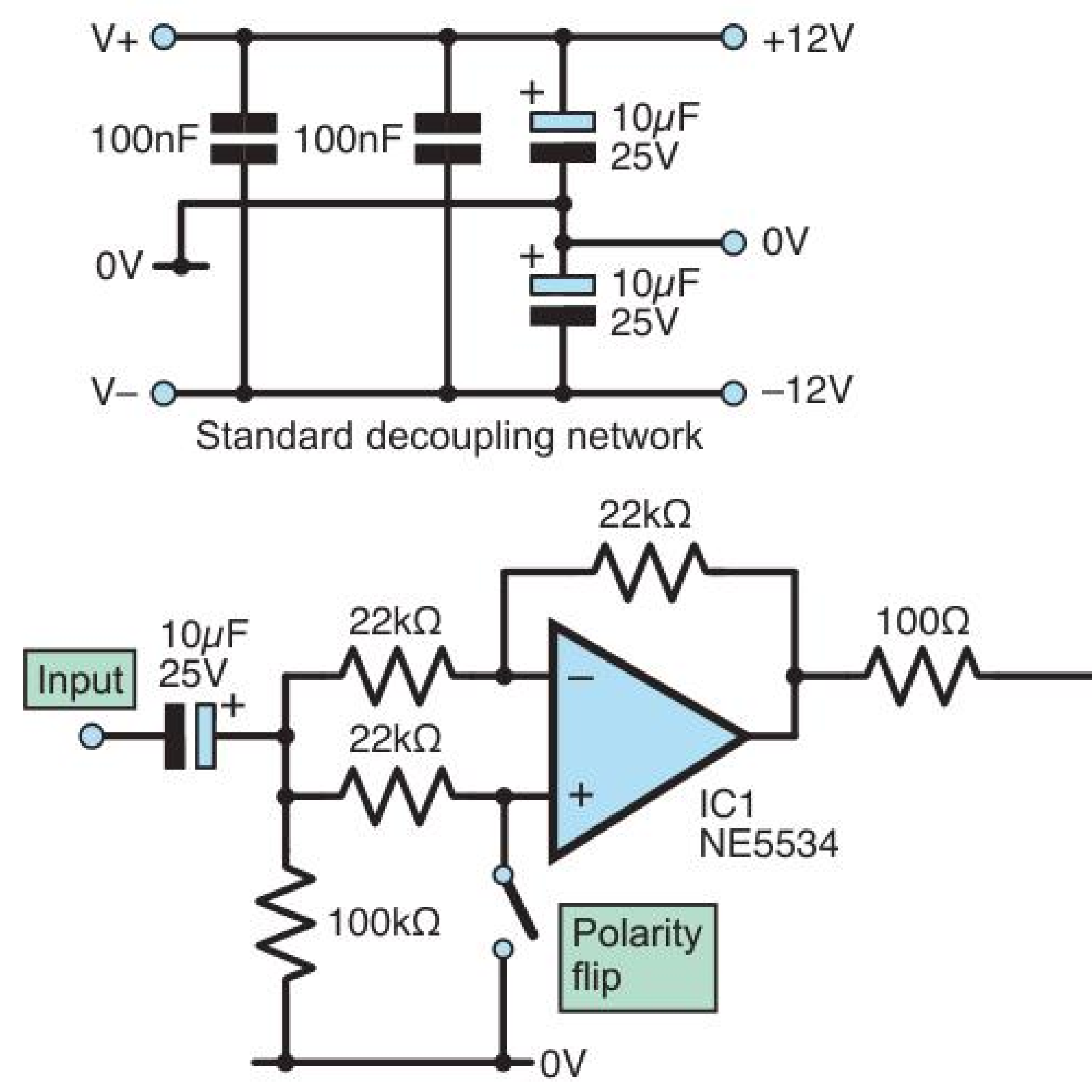
I don't approve of increased spacing, since the tweeter sounds disembodied when the system is used for near-field monitoring. In active speakers, it gives the best results to put the drivers' flanges virtually touching and then dial in the correct time alignment.

Photo 15: this speaker, based on Wavecor drivers (*PE* Feb-Mar 2020) has the tweeter next to the woofer to ensure good integration up-close. No all-pass filter was necessary due to an asymmetric crossover.

I have given up determining this mathematically, since I can get it right much quicker by twiddling a knob on the all-pass filter tweeter compensation jig I made, shown in Fig.29 and Photo 16. This will compensate for the variables, including the unknowns. When I get the perfect summation and reverse polarity cancellation dip, I simply measure the resistance value of the pot.

Of course, if you later alter the crossover filters or driver equalisation, it will need readjusting again. This is how loudspeaker design is for the home constructor, unless you have accurate drive unit models and love playing with complex software.

There are some excellent books on the computer-based design


V+
+12V
100nF
100nF
10µF 25V
0V
0V
10µF 25V
V–
–12V
Standard decoupling network
22kΩ
Input
10µF 25V
22kΩ
22kΩ
100Ω
IC1 NE5534
100kΩ
Polarity flip
0V

of loudspeakers, such as Vance Dickason's *Loudspeaker Design Cookbook*, and the software is getting cheaper. You don't have to spend £5k on proprietary software to run on a £500 computer anymore, but £375 software, such as Crosslite+, may be needed. (I've not tried it, only watched a video showing how to use it, and it looked 'too easy').

Watch out for subscriptions as well; they are becoming an unpleasant certainty in life, along with death and taxation.

Just to be contentious, I would say that with the most popular computer-designed speaker format, the LR fourth-order reflex, makes the speakers all have a similar sound, tailored for the studio showroom and possibly no better than good empirical designs such as the Spendor BC1 and BBC LS3/5a.

Conversely, most of the inaccurate speakers around seem to have been designed by people with defective hearing.

### Bass-to-mid crossover phase

The effect of a physical displacement in the drivers is usually insignificant at the lower bass/mid crossover points due to the long wavelengths involved.

Using the equation $\lambda = c \div f$, the "Ohm's Law" of acoustics, for say $f$ = 300Hz, (a common crossover point for 3-way designs with 100mm mid units), the wavelength $\lambda$ = 343m/s ÷ 300Hz = 1.14m.

A typical driver displacement distance is around 60mm, which is virtually insignificant compared to a wavelength of over one metre.

### Next month

We are running out of space but still have some unfinished business, so this topic will continue next month, including some more all-pass filter theory.

That will include some hybrid crossovers used by KEF that provided group delay for phase alignment as well as crossing over the drivers in the speaker.

After that, we'll look at combining regular and all-pass filters to make new kinds of filters, including those that use signal subtraction to form new filters. We'll also discuss the purpose of Linkwitz-Riley crossovers in a bit more detail.

After that, we'll get to the practical side of this topic and build a universal all-pass filter board, with up to eight cascaded all-pass sections or poles available, built around four dual op amp chips. **PE**

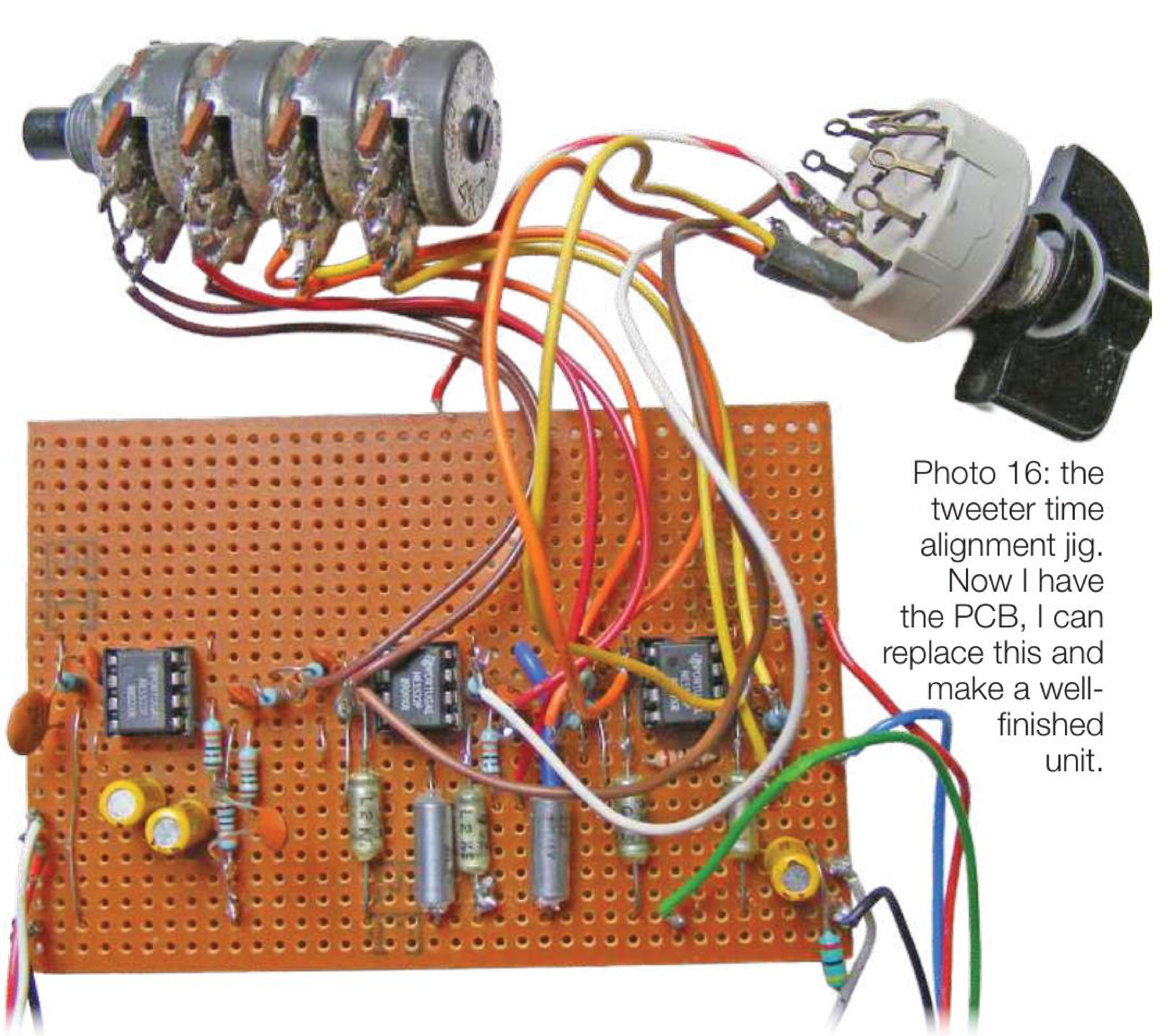


Photo 16: the tweeter time alignment jig. Now I have the PCB, I can replace this and make a well-finished unit.

5-year collections

2019-2023
£49.95

2018-2022
£49.95

2017-2021
£49.95

2016-2020
£44.95

Purchase and download at:
www.electronpublishing.com

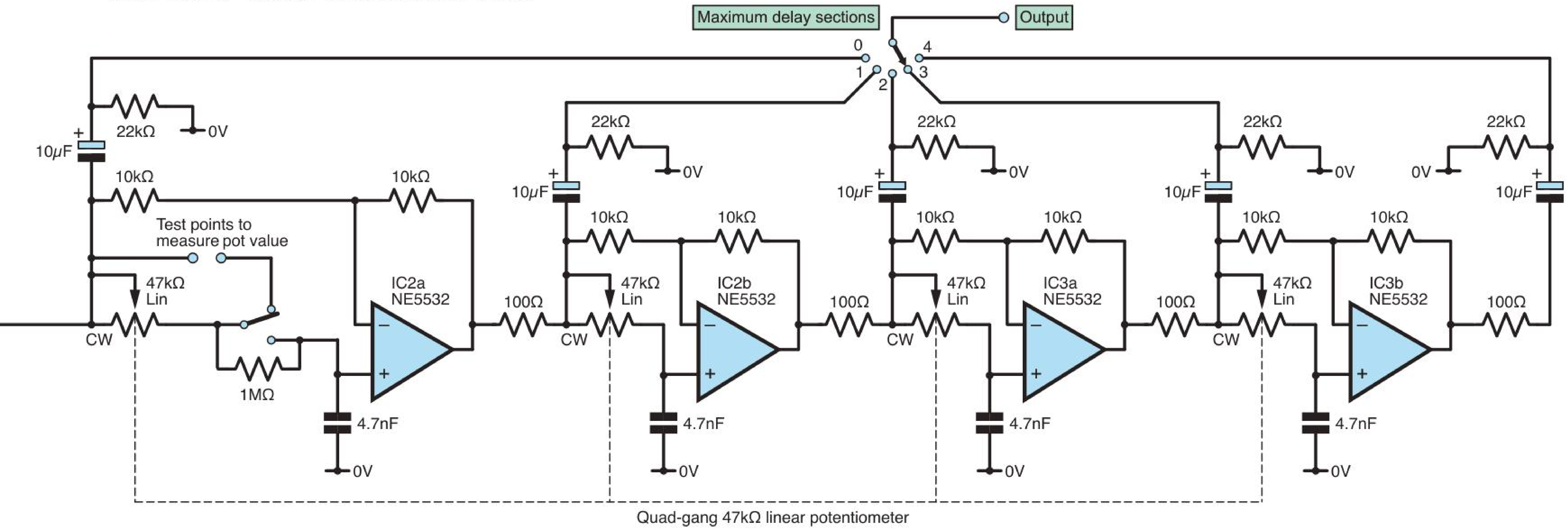


Fig.29: an adjustable all-pass filter for tweeter – a useful bit of kit for any active speaker designer. It enables you to find out the number of all-pass stages required and the optimum RC frequency by twiddling a knob.

### PROJECT BY PHIL PROSSER

# Tool Safety Timer

Safety
12V DC
START
RUNNING
PIR

**Have you ever accidentally left your soldering iron switched on? This project will help allay your fears. It's also a great idea for fitting to a hot air rework system or other mains-powered tools.**

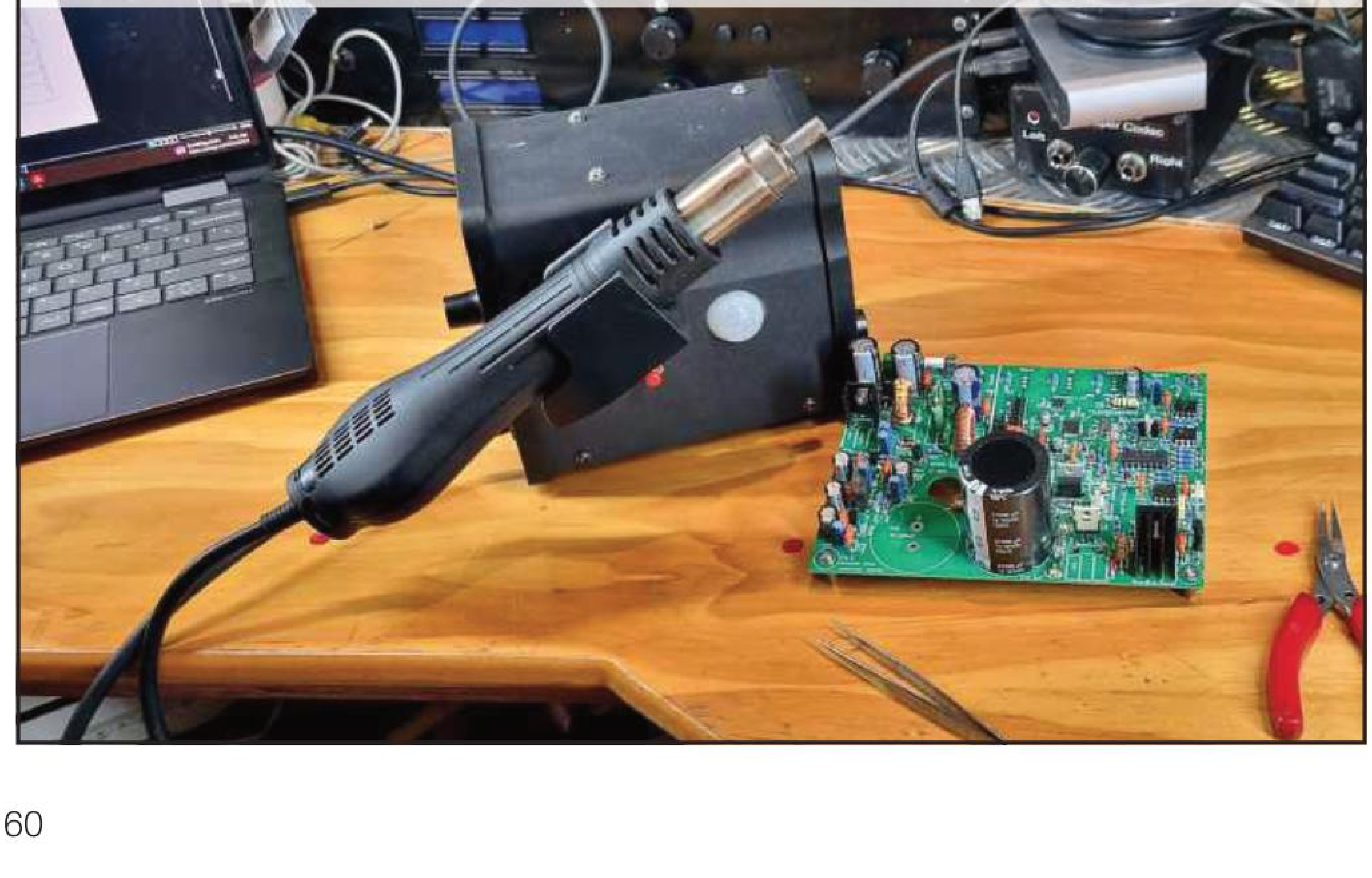
- Switches a mains-powered device off after a period of inactivity
- Activity of user movement sensed using a PIR (passive infrared) sensor
- Selectable timeout of 10, 20, 30 or 60 minutes
- Pushbutton to switch tool back on again after timeout
- Indicator LED shows when PIR is picking up motion
- Power supply: 12-20V DC, 9-15V AC @ 200mA maximum with the device handling up to 10A/2.3kW

**The hot air gun** we use is one of those ubiquitous ones sold under many different brands. These tools are quite low in cost but function surprisingly well. To operate them, you lift the handpiece and the heater and fan start automatically. They run until you place the handpiece back in the cradle, which is sensed by a magnet in the cradle and everything cools down.

The hot air gun is often set well in excess of 300°C and is therefore fully capable of igniting combustible material. Unfortunately, this author is really forgetful and often comes back hours later to find the hot air gun ready to go. If, for example, a mischievous cat brushes against it, or a child decides to play with it, the hot air gun may fall from its cradle with nobody around to avert disaster.

This has the real potential to set the house on fire, which is the one thing about this tool that we really dislike. It should shut itself down after a period of being idle.

On a somewhat less dramatic note, our soldering iron is left on more often than not. The one in our lab does not have an automatic shutdown, and we have found it running at full temperature after many days. This is much less likely to cause a fire, but we have replaced many more tips than necessary (and wasted expensive electricity too).



**My modified hot air gun. We mounted the PCB to the inside of the lid, a transformer to the base and the sensor and start switch to the side. This has proven to be very effective. Yes, our hot air gun has seen some use.**

This project is to provide a very simple and convenient approach to automatically switching these tools off. You run your tool from a switched mains source with a timer you can set. If you are present and moving the tool around, or moving around yourself, the timer will be reset. If you leave the room, when the timer expires, your tool is switched off.

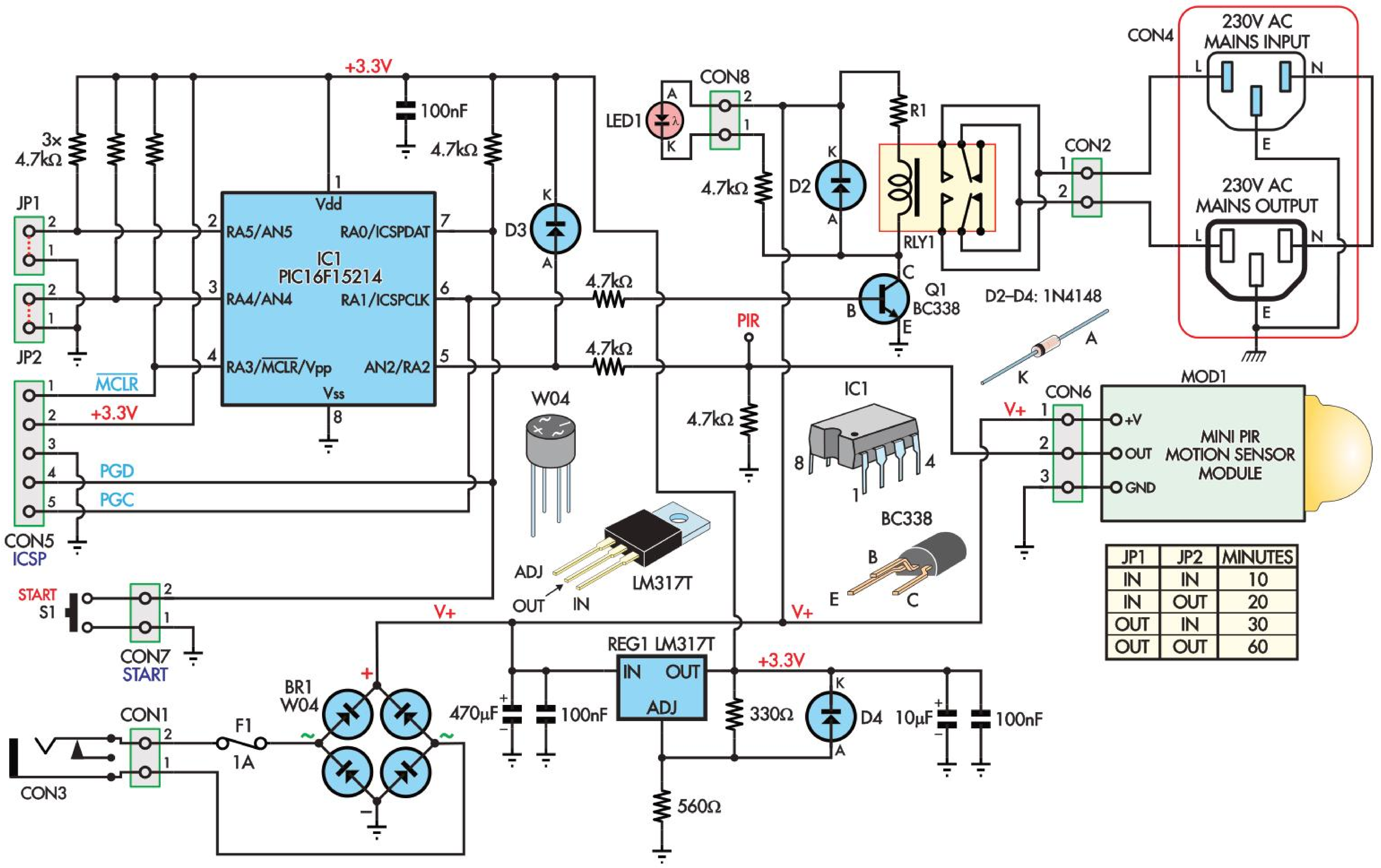
When you come back later, you just need to press a button on the box to restart it.

You might be sitting there for an hour or two and the tool might run for this time, but our theory is that if you are present, you would catch any significant hazard. Once you leave, the system will 'time out' and protect you and your tools.

Our tests have shown that with the sensor on the desk near the user and tool, it will easily detect the user's hand, and especially a hot soldering iron or hot air gun.

We recommended the PIR is pointed in the direction of your tool stand, ideally away from yourself and definitely away from thoroughfares. This way, it can detect when you stop using the tool or when you leave the desk. In practice, this works as long as there is a line-of-sight from the sensor to you or the tool when you place it in its stand.

We considered two ways to use the Tool Timer. The first is to mount it in a UB1 ABS plastic case and power it from an external 12V DC plugpack.


+3.3V
100nF
3× 4.7kΩ
4.7kΩ
JP1
JP2
1
2
3
4
5
6
7
8
Vdd
IC1
PIC16F15214
RA5/AN5
RA0/ICSPDAT
RA4/AN4
RA1/ICSPCLK
RA3/MCLR/Vpp
AN2/RA2
Vss
MCLR
+3.3V
PGD
PGC
CON5
ICSP
START
S1
CON7
START
D3
K
A
LED1
CON8
4.7kΩ
D2
R1
RLY1
CON2
CON4
230V AC MAINS INPUT
230V AC MAINS OUTPUT
L
N
E
Q1
BC338
B
C
E
D2–D4: 1N4148
PIR
4.7kΩ
4.7kΩ
4.7kΩ
W04
IC1
BC338
LM317T
ADJ
OUT
IN
MOD1
CON6
V+
+V
OUT
GND
MINI PIR MOTION SENSOR MODULE
V+
V+
REG1 LM317T
IN
OUT
ADJ
+3.3V
BR1
W04
+
~
–
CON1
CON3
F1
1A
470µF
100nF
330Ω
D4
10µF
100nF
560Ω

| JP1 | JP2 | MINUTES |
|---|---|---|
| IN | IN | 10 |
| IN | OUT | 20 |
| OUT | IN | 30 |
| OUT | OUT | 60 |

**Fig.1: the circuit is based around microcontroller IC1 and the PIR sensor. When the sensor's output goes high, meaning it has detected movement, the timer is reset. Once the time period has elapsed, the relay is de-energised, switching off the tool. It can be switched on again by pressing S1.**

The second is to embed this in a tool, like our hot air gun. We will describe how to package it in the UB1 case in detail.

Because we do not know the specifics of your tools, we can't go into great detail on how to fit it inside a tool. However, we will provide notes and advice on this. If you are not completely confident and comfortable analysing and understanding how your specific tool operates, we recommend you stick to the UB1-packaged version.

## Circuit details

The Tool Timer circuit is shown in Fig.1. Switching of the tool power is via a 10A mains-rated relay. The recommended relay can actually carry and break 16A, which is more than enough. We have heavy tracks on both sides of the PCB so it can handle 10A continuously.

The Tool Timer switches the Live line only. Never assume that a device that is disabled by it is isolated from the mains.

We use an IEC plug/socket combo for the mains input and output. This has the Earth link integrated. We link Neutral to Neutral directly on this part, and wire the Live input through our PCB to the Live output. It does not matter which pins of CON2 are used as the input and output.

For the PIR sensor, we chose one that can operate from a wide range of voltages, up to 20V DC, and provides a 3.3V logic high level on the middle pin when an object is detected. We have included a test point on the PCB to which you could connect another LED with a series resistor to indicate when an object is being detected by the PIR. This is labelled "PIR".

This PIR signal is fed to the PIC and, if movement is detected before the Tool Timer has switched the load off, it will reset the timer. This means that if you use a soldering iron or heat gun, taking the tool from its holder and replacing it, the PIR will sense this and reset the timer. So it will never time out while you are actively working.

The PIC has two jumpers, JP1 and JP2, that allow you to program a timeout period of 10, 20, 30 or 60 minutes. The timer starts when the PIC is powered up, and the relay is switched on while the timer is counting down. In practice, it won't expire if you are using the tool; there is even time to get a cup of coffee, even from the shop, if you are using 30 or 60 minute timeout!

Once the timer expires, the relay switches off, and will only switch on again if you press the START button on the front panel (S1). That is really all there is to this project.

Other uses of the Tool Timer are quite varied. It could be used as a night light that times out after 60 minutes and after people stop moving in the room.

Another potential use is as a timer for kitchen equipment, such as coffee makers, toaster ovens etc. You can start the device by pressing the button and it will continue to run while people are around and for a set period thereafter. This will help avoid that burned coffee smell when the pot is left running too long, or that toasty maker sitting there running forever, waiting to burn unsuspecting fingers.

## Power supply

The power supply includes provision to mount a bridge rectifier, which might seem odd. The intent is that if you are integrating this into an existing piece of equipment, it may be easier to power it from AC. Our heat gun, for example, has 9V AC accessible. With the bridge rectifier, this works just fine.

Of course, you can run DC via the bridge rectifier, provided you have at least 12V available for driving the relay.

We use an LM317 regulator to generate 3.3V for the PIC, but we operate the PIR from the raw DC. The recommended PIR has its own regulator and can accept 5-20V DC. This same PIR sensor module is widely available from various suppliers on the internet, so it should not be difficult to get.

## PCB design

We have added slots in the PCB to ensure isolation between the mains and low-voltage sections of the board. They will make certain that whatever load you connect won't cause arcing across the relay tracks.

## Construction

The Tool Timer is built on a double-sided board coded 10104251 that measures 71 × 88mm. Assembly is fairly easy as all the parts are through-hole types and the board is not tightly packed. Start with the resistors; fit all the 4.7kΩ parts, then the others. Next, mount the three 1N4148 diodes, making sure they face as shown in the PCB overlay diagram, Fig.2.

We have specified a W02/W04 (or similar) diode bridge. If you are running this from DC, you could install wire links in its place, as shown on the PCB and in Fig.2. In our testing, using a 12V DC plugpack, the relay operates just fine with this bridge in place, but it is not necessary.

Double-check that you are inserting it the right way around (pay attention to the markings on the device and the board), as these are really fiddly to remove from the PCB if installed incorrectly.

Follow by fitting the fuse clips (with the retention tabs on the outside) and all the capacitors, watching the polarity of the electrolytics. Next, mount the headers and screw terminal. The screw terminal specified is rated to 300V AC and 16A. This part has a standard 5.08mm lead spacing; if you substitute it, make sure the part is rated for at least 250V AC and the current you are switching.

Now mount the transistor, relay and regulator. The relay has a very standard footprint and you will find many options. If you substitute this part, again you need to pay attention first to the voltage and current ratings, as well as the coil voltage rating.

We have wired the relay to operate from the input supply. If this is a 12V DC plugpack, we want this to power the relay directly. In this case, R1 is a wire link (ie, 0Ω). If your DC supply is more than 12V, we want to drop this back to 12V with a resistor.

This dropper does not need to be very accurate. If the relay gets a supply within a volt or two of 12V, it will be OK.

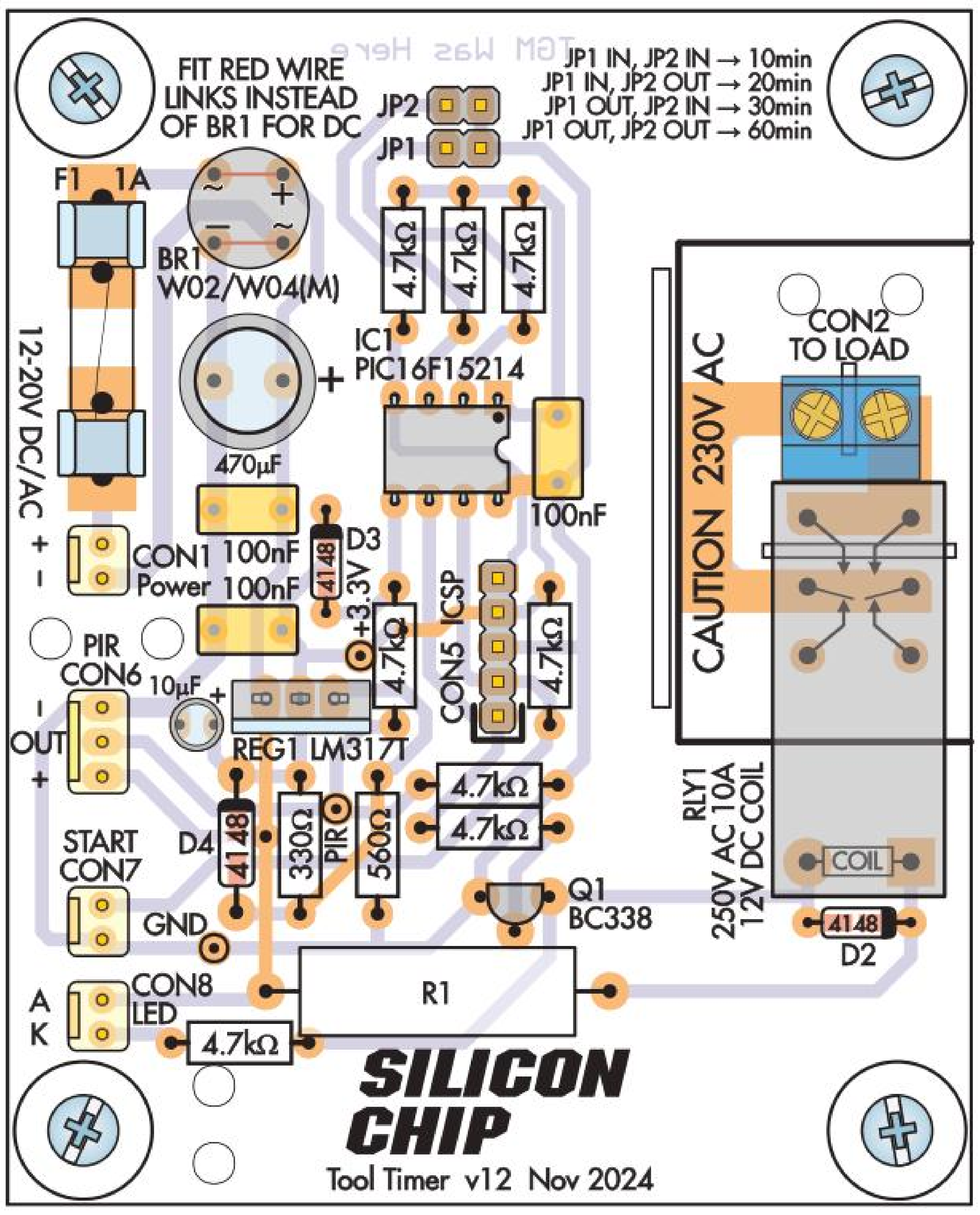


**Fig.2: there is nothing terribly difficult about assembling this PCB. We don't recommend using a socket for IC1, as it could fall out and cause a safety hazard. Watch the orientation of the bridge rectifier, electrolytic capacitors, regulator, diode and IC and note the three safety clearance slots.**

### Table 1 – R1 values for a DC supply (BR1 linked out)

| Supply | R1 value |
|---|---|
| 12V | 0Ω (wire link) |
| 14V | 33Ω ¼W |
| 16V | 68Ω ½W |
| 18V | 100Ω 1W |
| 20V | 150Ω 1W |

### Table 2 – R1 values for an AC supply (BR1 loaded)

| Supply | R1 value |
|---|---|
| 9V | 0Ω (wire link) |
| 11V | 39Ω ¼W |
| 13V | 82Ω ½W |
| 15V | 150Ω 1W |

Table 1 provides values for R1 if your DC supply us higher than 12V, while Table 2 provides various R1 values for AC supplies. If using an AC supply, it must not exceed 15V.

With all the parts aside from the PIC loaded, apply power to your board using a plugpack or bench supply and measure the voltage between pins 1 and 8 of IC1's pads. The reading should be 3.3V ±0.2V. If this is not right, check that the bridge is in the right way around and that the regulator is mounted correctly. Also check the 330Ω and 560Ω resistors and diodes.

With the power supply operating OK, disconnect the power supply and fit the PIC. Double-check its orientation before soldering it. If yours is preprogrammed, you can move onto the Packaging section. There is a programming header on the PCB (CON5); this uses the standard Microchip pinout so a SNAP or PIC kit can be plugged straight in. You can download the required firmware from siliconchip.au/Shop/6/1825

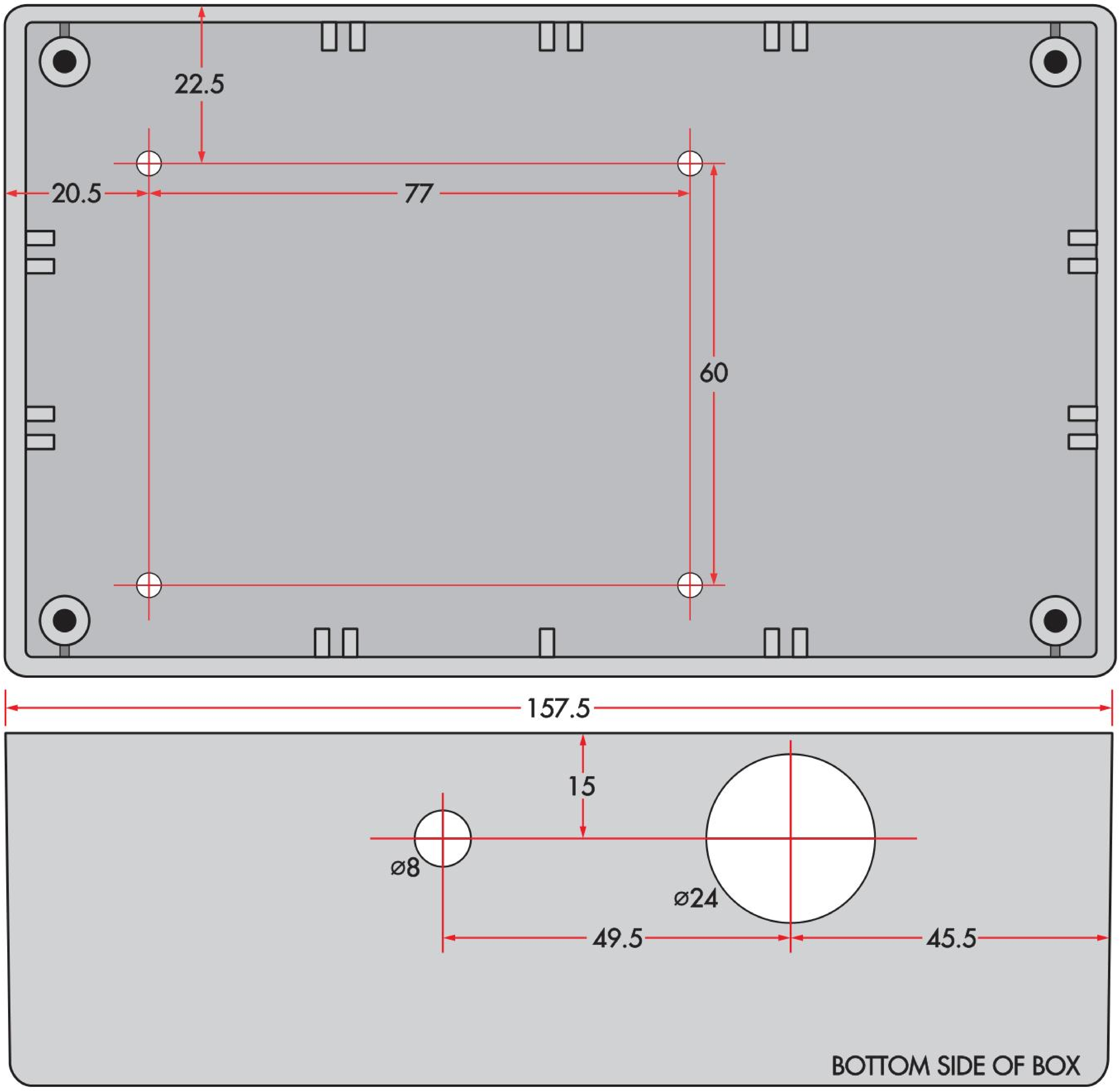
## Packaging

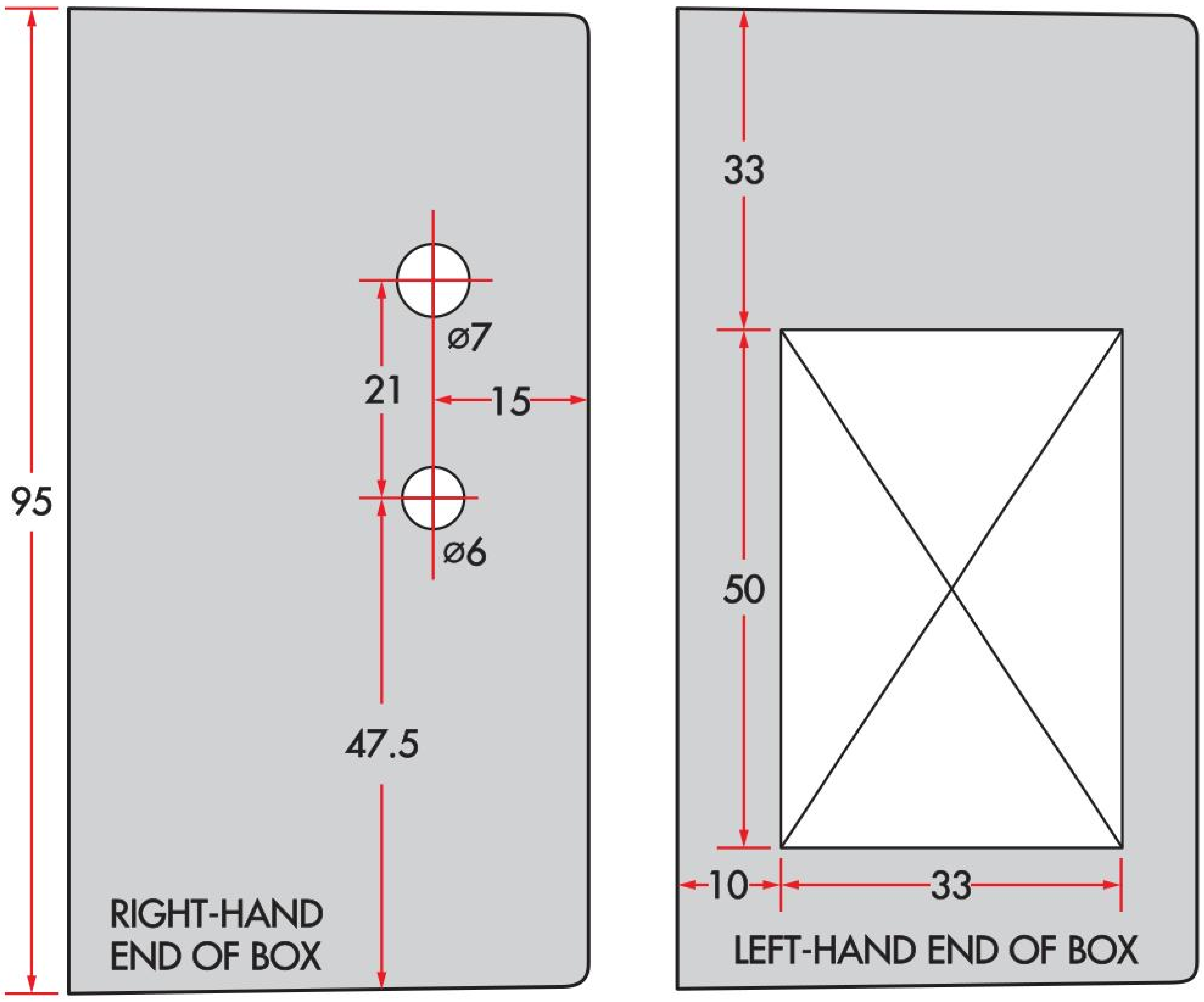
A UB1 Jiffy box is a good match for this board and associated parts. Figs.3 & 4 show the holes and cutouts that are required. Mark and drill the round holes first. We used a stepped drill for the PIR hole. If you have never used one of these, we reckon you should try one – they are awesome for making larger holes in plastic and aluminium.

There are a couple of locations where you will need to trim back the PCB guide rails on the inside of the case using a sharp knife or chisel.

For the holes in the base for the PCB, a quick cheat is to drop the board into the case and mark through the mounting holes. We wanted the PCB on the opposite side of the box from the PIR to allow room for connectors, and we suggest you do the same.

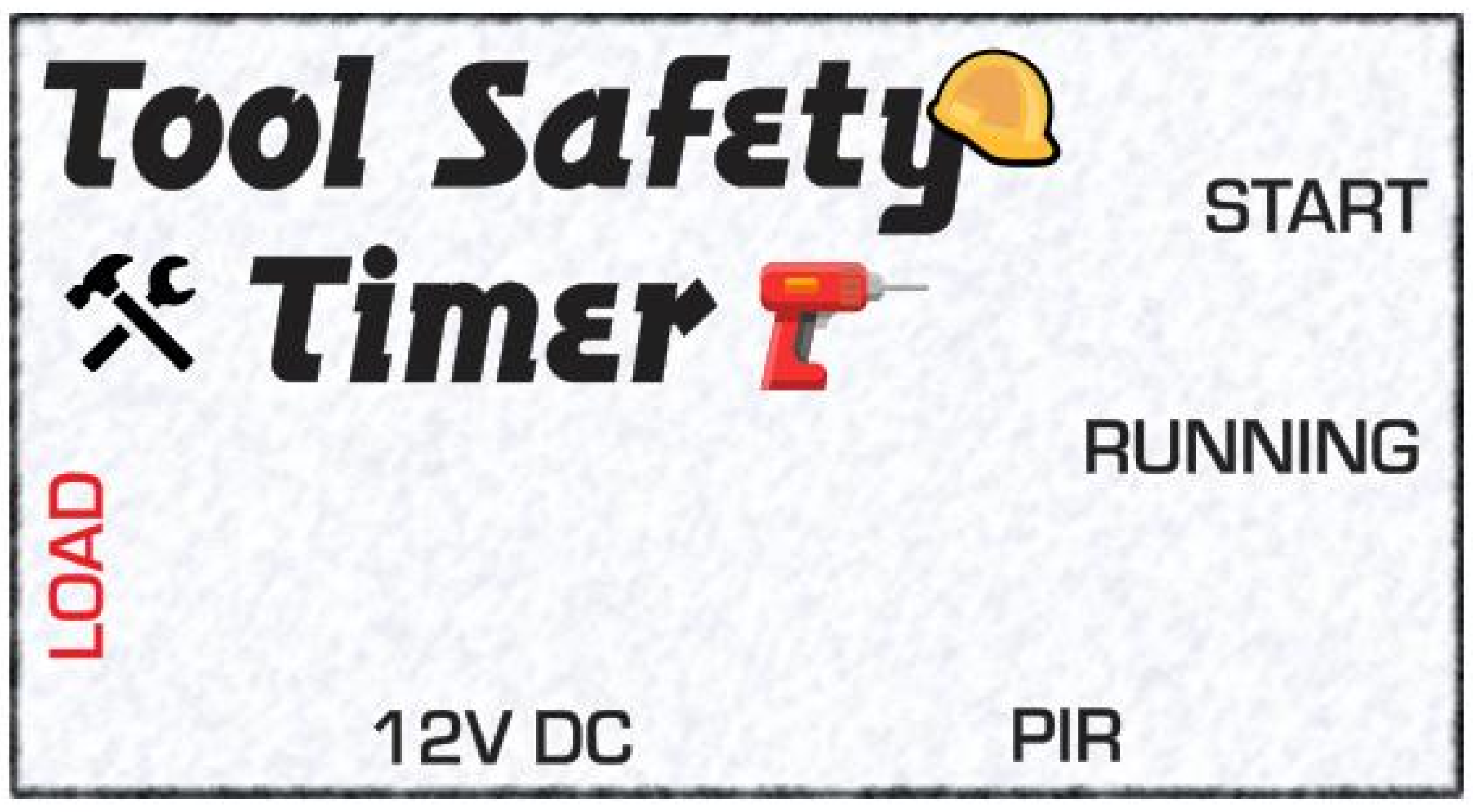
We used an oscillating multi-tool for the rectangular cutout, which made a


22.5
20.5
77
60
157.5
15
ø8
ø24
49.5
45.5
BOTTOM SIDE OF BOX


95
21
15
ø7
ø6
47.5
RIGHT-HAND END OF BOX
33
50
10
33
LEFT-HAND END OF BOX

**Figs.3 & 4: just four holes are required in the base – you can use the PCB as a template to mark their positions before drilling. Make the holes as shown here (and in Fig.6 overleaf) to verify how they relate to each other. You can use a multitool or rotary tool to cut the rectangular opening, or drill a series of small holes inside its perimeter and then file it to shape.**

**Fig.5: the label for the Tool Safety Timer is shown here at 50% actual size. You can download it from siliconchip.au/Shop/11/1827**


Tool Safety
Timer
START
RUNNING
LOAD
12V DC
PIR

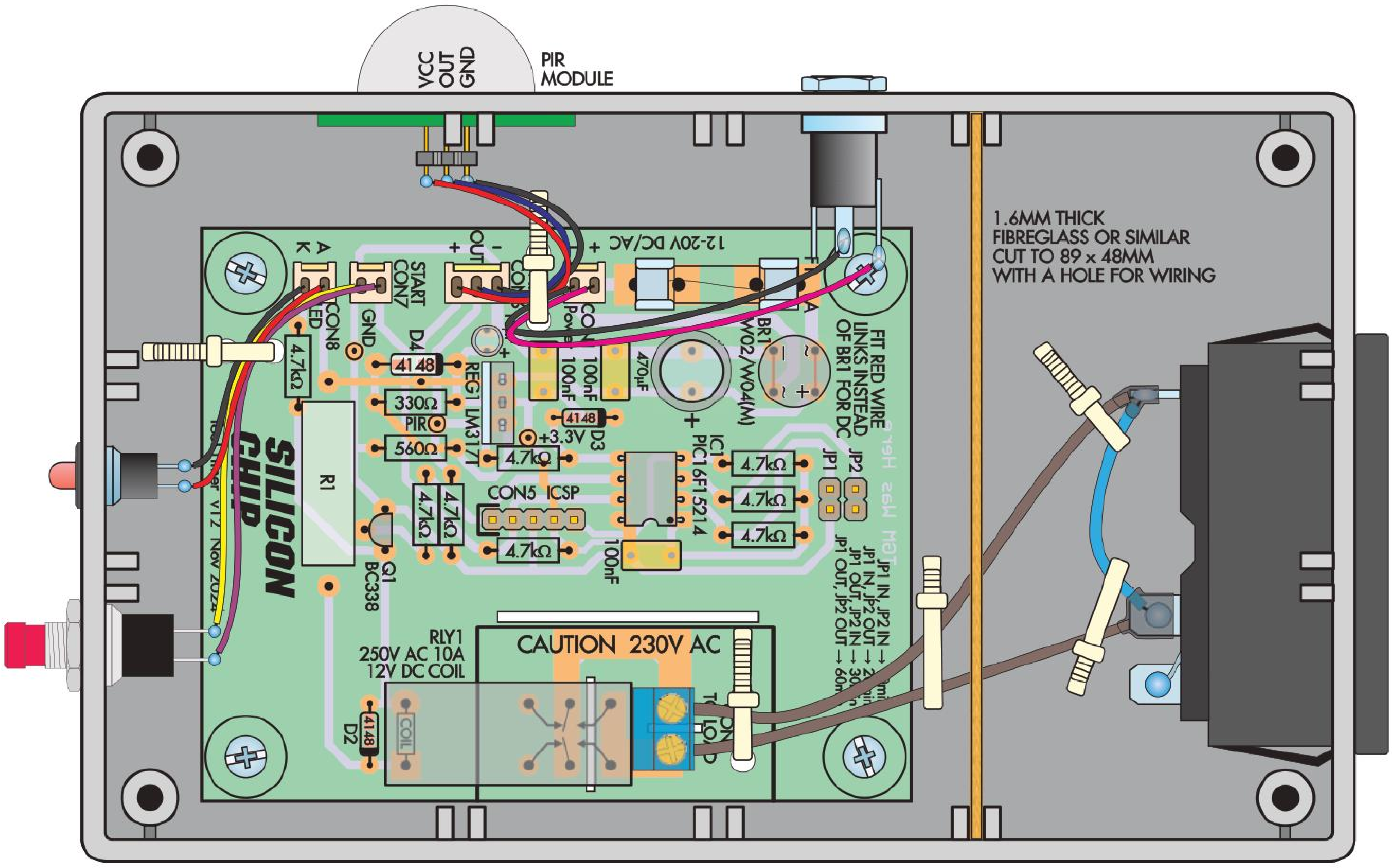
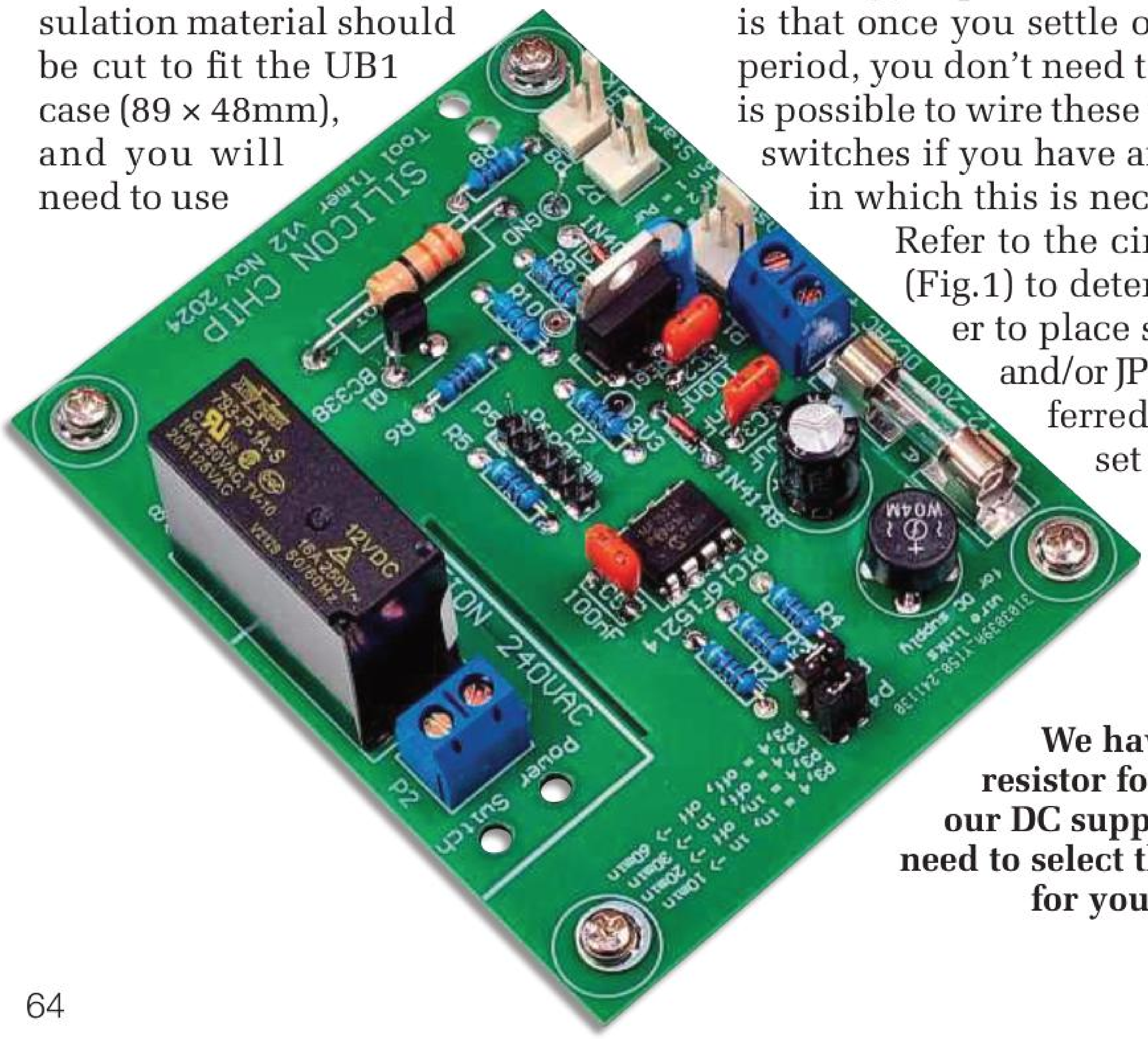

VCC
OUT
GND
PIR MODULE
1.6MM THICK FIBREGLASS OR SIMILAR CUT TO 89 x 48MM WITH A HOLE FOR WIRING
12-20V DC/AC
CON1 Power
START CON7
CON8 LED
A
K
GND
D4 4148
330Ω
PIR
560Ω
REG1 LM317T
+3.3V
4.7kΩ
100nF
470µF
4148 D3
BR1 W02/W04(M)
FIT RED WIRE LINKS INSTEAD OF BR1 FOR DC
IC1 PIC16F15214
CON5 ICSP
JP1
JP2
R1
Q1 BC338
SILICON CHIP
RLY1 250V AC 10A 12V DC COIL
CAUTION 230V AC
COIL
4148 D2

**Fig.6: here is how to run the wiring. While you could solder most of the wires to the PCB, the headers and plugs make it much easier to disassemble it should you need to. Don't omit the cable ties and make sure the mains wires are correctly rated, the right colours and routed to the appropriate terminals (they are usually marked L (or A), N & E on the connector).**

somewhat fiddly job easy. So if you have one, crack it out for this part of the job. The recommended case is made of ABS plastic, which is not hard, so you can easily use a small hand saw for this. Make sure this hole starts smaller than needed and file it to the final size.

Lastly, you should add a 1.6mm thickness (or more) piece of insulating material like fibreglass between the IEC socket and PCB. The insulation material should be cut to fit the UB1 case (89 × 48mm), and you will need to use a sharp knife to cut a small hole to allow wiring to pass through.



**We have used a 33Ω resistor for the relay, as our DC supply is 14V. You need to select the right value for your application.**

## Setup and wiring

With the box ready, we can now start pulling it all together. Use the wiring diagram, Fig.6, and photo opposite as guides while you read the following steps.

Headers JP1 and JP2 set the timeout period to 10, 20, 30 or 60 minutes. We are using jumpers since our experience is that once you settle on a workable period, you don't need to change it. It is possible to wire these to a switch or switches if you have an application in which this is necessary.

Refer to the circuit diagram (Fig.1) to determine whether to place shunts on JP1 and/or JP2 for your preferred timeout and set that up now.

Next, complete the wiring to the panel-mounting LED, start button, PIR sensor and power input connector. You need these installed in the case and secured with the appropriate washers and nuts before you connect the headers to their connecting wires, as the headers will not fit through the holes in the case.

Assuming your DC supply is from a plugpack and you'll be using a barrel socket, wire it to the board using two 90mm lengths of light-duty hookup wire. We used red and black, with the red wire going to the positive middle pin on our 2.1mm inner diameter barrel socket.

For a DC supply, make sure that your positive wire goes to the pin of CON1 marked +. If you don't have the crimping tool for the plug, you can use a pair of side cutters or needle-nose pliers to crimp the wire in. This crimp will not be great, so solder over your temporary crimp, and it will be secure. Just don't add a lot of solder or you won't be able to insert the pin into the plastic block.

For the PIR, use 70mm of light-duty wire. We used a ribbon cable offcut as this helps keep the wiring tidy. You need to ensure that the headers are wired correctly, ie, the + pin on the PCB goes to the positive supply on the module, the – pin goes to the negative supply and the centre pin goes to the PIR output.

For the LED, use two 90mm lengths of light-duty hookup wire or another ribbon cable offcut. You need to make sure you get the anode and cathode to the right pins. The cathode has a chamfer on the side of the LED and a shorter pin (it goes to the pad marked K).

For the start button, use another two 90mm lengths of light-duty hookup wire or a ribbon cable offcut. Any colour will do. The button is not polarised, so connect the button to the two-way plug however you want.

## Testing

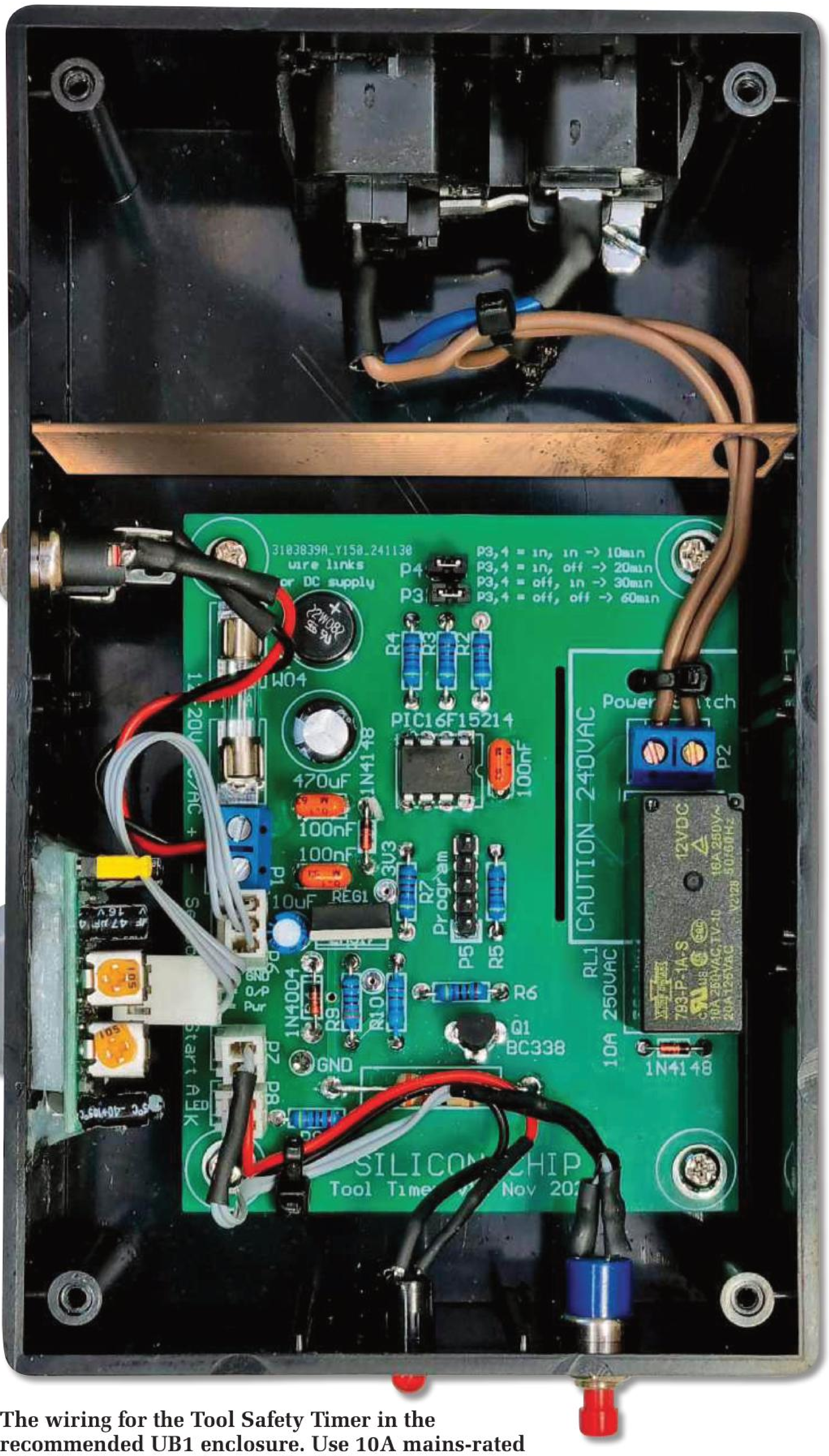
With the timer set to 10 minutes (JP1 & JP2 in), apply power. Put something over the PIR sensor so it cannot detect your presence; a sheet of paper will do the trick. The LED should light and stay lit for pretty close to 10 minutes, then it should switch off. You should hear the relay switching along with the LED.

If the LED stays on permanently, check that you have the PIR covered and that it is not detecting you move around, as it will reset the timer and the LED won't go off. Also check that the voltages on the PIR are right. You should be able to monitor the PIR output using a DVM on the PCB test point, and see the PIR detecting your hand if you wave it in front of the PIR.

If it still isn't working, check for short circuits on the PCB around the PIC and the PIR header, and verify that the transistor is in the right way around and that you have used an NPN type (BC337 or BC338).

Once it has timed out, press the start button. The relay should switch and the LED should come on again for a further 10 minutes.

Finally, run the same test with the PIR pointing in your general direction and wait the 10 minutes. Unless you sit statue still, the PIR will sense you and the LED and relay should stay on.



The wiring for the Tool Safety Timer in the recommended UB1 enclosure. Use 10A mains-rated wire for the connections to the IEC connector.

## Final assembly

We can now assemble this lot into the enclosure. Install the standoffs using shakeproof washers and machine screws, then mount the PCB into the enclosure. If you haven't already, plug the power, start, PIR and LED headers onto the board.

Now you can zip tie both the PIR and power wires to the PCB using the two holes provided next to the power connectors. Similarly, zip tie the start button and LED wires using the lower set of 3mm holes provided on the PCB. We recommend applying a drop of Loctite or similar glue to the connectors to secure them to the board.

Now install the IEC connector into the enclosure. Ensure it is secure and the mounting snap-in tabs hold it in place. If it is not totally secure, fix that before proceeding.

## Parts List – Tool Safety Timer

1 double-sided PCB coded 10104251, 71 × 88mm
1 UB1 Jiffy box, 158 × 95 × 53mm
1 1.6mm-thick insulation material cut to 89 × 48mm (fibreglass, acrylic, Presspahn or similar)
1 12V DC 200mA+ power supply (eg, plugpack)
1 12V DC coil 250VAC/10A+ SPST PCB-mount relay (RLY1) **[eg, SMI-S-112LM (AliExpress 1005004502346634]**
1 PIR motion sensor module (MOD1) **[HC-SR501, AliExpress 1005006824167570]**
2 M205 PCB-mounting fuse clips (F1)
1 M205 1A fast-blow fuse (F1)
1 SPST red panel-mount pushbutton switch (S1)
2 2-way headers, 2.54mm pitch (JP1, JP2)
2 jumper shunts (JP1, JP2)

**Connectors**

3 2-pin polarised headers with matching plugs and pins (CON1, CON7 & CON8)
1 2-way mini terminal block, 5/5.08mm pitch (CON2)
1 chassis-mounting barrel socket (to suit power supply) (CON3)
1 10A 250V IEC mains power input (C13) & output (C14) socket combination (CON4)
1 5-pin header, 2.54mm pitch (CON5; **optional** – for ICSP)
1 3-pin polarised header with matching plug and pins (CON6)

**Hardware, cable & wire**

8 M3 × 6mm panhead machine screws
8 M3 shakeproof washers
4 M3 × 10mm tapped spacers
7 100 × 2.5mm Nylon cable ties
1 100mm length of 5mm diameter black heatshrink tubing
1 100mm length of light blue 10A mains-rated wire
1 300mm length of brown 10A mains-rated wire
1 IEC C13-C14 mains extension cable
4 stick-on rubber feet

**Semiconductors**

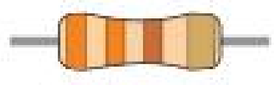
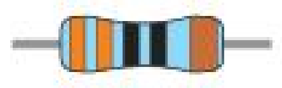
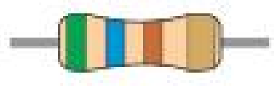
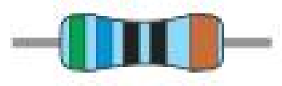
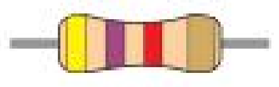
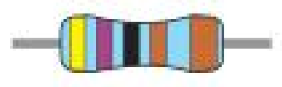
1 PIC16F15214-I/P programmed with 1010425A.HEX, DIP-8 (IC1)
1 LM317T 1A adjustable regulator, TO-220 (REG1)
1 BC337 or BC338 25V 800mA NPN transistor, TO-92 (Q1)
1 red 5mm LED with bezel (LED1)
1 W02(M) or W04(M) 1.2A bridge rectifier (BR1)
3 1N4148 75V 200mA signal diodes (D2-D4)

**Capacitors**

1 470µF 35V electrolytic
1 10µF 50V electrolytic
3 100nF 63V MKT

**Resistors (all ¼W 1% axial unless noted)**

8 4.7kΩ
1 560Ω
1 330Ω
1 0-150Ω 0.25-1W resistor (R1; see Table 1 & text)

| Qty | Value | 4-band code | 5-band code |
|---|---|---|---|
| 1 | 330Ω |  |  |
| 1 | 560Ω |  |  |
| 1 | 4.7kΩ |  |  |

Check that the Earth link is good (that the two Earth pins in the IEC socket have a very low resistance between them). It is part of the connector and it's critical for safety, so we want to make sure it is OK. You will need to solder the Earth link on the back of the IEC connector as its fixing assumes a wire will be soldered to it.

Now connect a short length of 10A light blue mains-rated wire between the terminals labelled N on the rear of the connector.

Use two pieces of 5mm diameter heatshrink tubing to insulate the connections of this wire to the IEC connector tabs. Note that this wire will cross over the Earth strap.

Now take two 150mm lengths of brown 10A mains-rated wire and connect them to the Live terminals on the IEC connector. Insulate these with 5mm heatshrink tubing too. Run a 2.5mm cable tie around the Live and Neutral wires, under the ground strap, and secure them together.

With the PCB in the case, trim the length of the two brown Live wires so they neatly present to the load switch connector, CON2, on the PCB. Once wired to the terminal, secure it to the PCB using a cable tie.

Stick on four rubber feet onto the case so you don't scratch your desk, and the Tool Timer is complete.

Finally, you can attach a label to the lid. This is a simple and cheap way of making this utilitarian project that little bit neater. The label can be downloaded from **siliconchip.au/Shop/11/1827** and you can find instructions for printing and attaching it at **siliconchip.com.au/Help/FrontPanels**

We used a 0.75m IEC mains extension cable to connect our soldering station to the tool timer. In operation, simply press "Start" on the tool timer and, while you are around and moving, your tool will stay on. When you wander off, it will switch itself off. Simple!

We will not go into detail regarding how the tool timer can be installed inside a heat gun as there are too many inconsistencies in how these are built for instructions to be safe. Only attempt to do this if you thoroughly understand what you are doing.

After reverse engineering our hot air gun, we concluded the safest approach would be to shut down the entire controller after the timeout. We chose to install a small 12.6V transformer in the case alongside the main transformer to power the Timer. We then used the Timer's onboard relay to disable the main transformer that ran the hot air gun controller.

Now I am less worried about my thoughtless cat setting my house on fire!

## Modifications

If you want a timeout period different from the four options we have provided, the source code for the PIC firmware is available to download from **siliconchip.com.au/Shop/6/1825**

The definitions for the timeout period are in the header file "util.h", defined in four lines (starting with #define Time_10Min_Runtime).

The timer counts in tenth of a second intervals, so 10 minutes is 6000 counts. Hence, each value is the required number of seconds multiplied by ten. PE

# Boy, that escalated quickly

Techno Talk

Max the Magnificent

From humble 8-bit microprocessors and Arduino boards to chiplets, XPUs, stacked silicon and edge-AI accelerators, a ramble through the astonishing (and slightly bewildering) evolution of modern computing.

**I'm starting to feel like an old** fool. Things are moving so fast these days that I don't know whether I'm coming or… I'm sorry, what were we talking about?

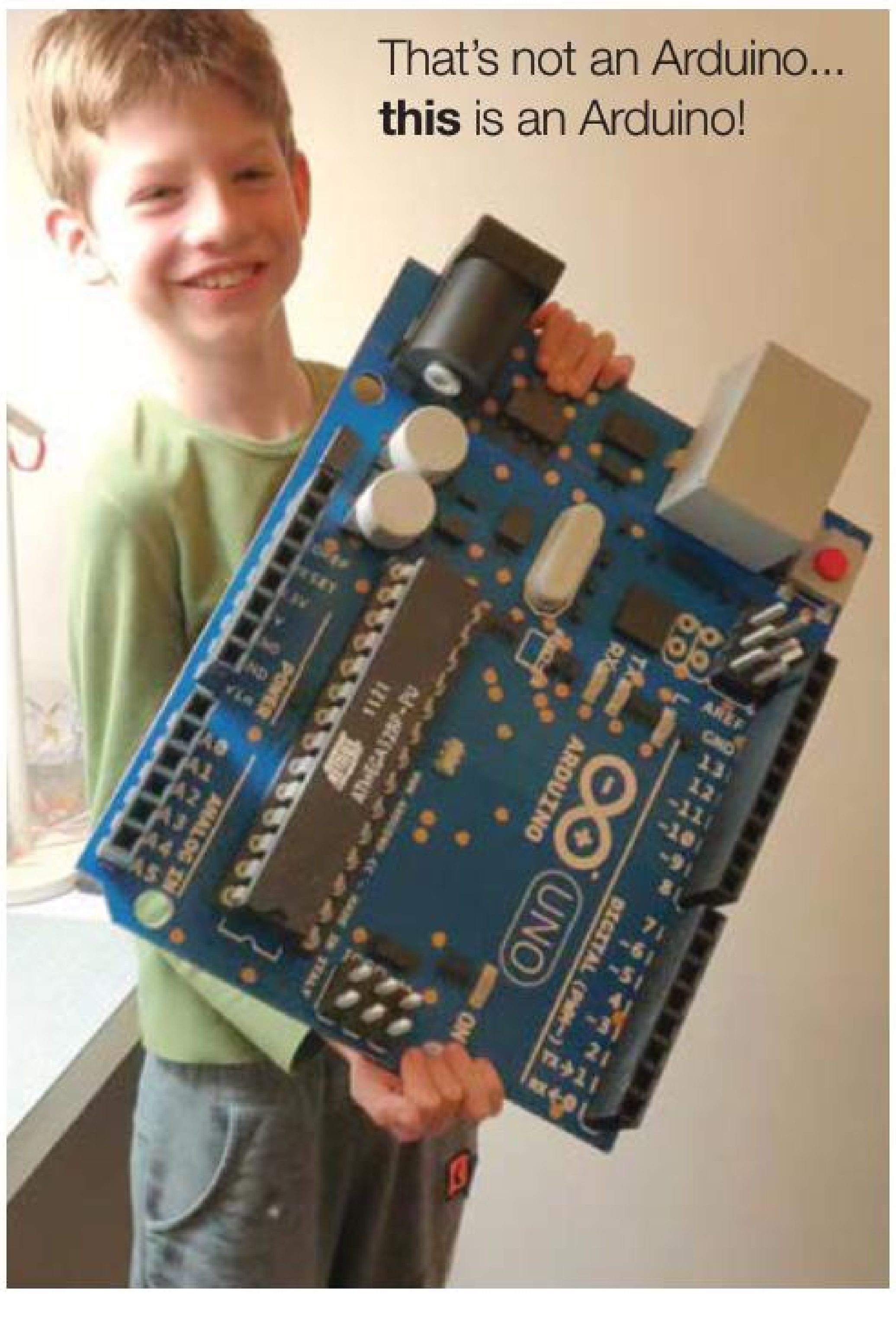
## Hello Arduino, my old friend

As I've mentioned on many an occasion (and as I'll doubtless mention again), I hail from the world of early 8-bit processors—those far-off days when a 1MHz system clock seemed scary fast.

Just to set the scene, typical high-end microprocessors in the mid-1970s, such as the Intel 8080 and Motorola 6800 (both announced in 1974), plus the MOS Technology 6502 (introduced in 1975), all had 8-bit data paths and clocked at about 1-2MHz. All required external RAM, external ROM, external clock circuitry, and external input/output (I/O) chips. If you needed an analog-to-digital converter (ADC), that would be external as well.

It's worth remembering that the introductory prices for the 8080 and 6800 were $300+ (~£150) and $200+ (~£100), respectively (that's $1800/£1450+ and $1200/£950+ in today's money). After that, you still needed to splash the cash for external memory devices and support logic! So it isn't surprising that when the 6502 appeared on the scene at just $25/£10 (about $150/£120 today), it sent shockwaves through the industry.

Compare this to today's Arduino microcontroller development boards. I typically use Arduino Uno or Nano boards for my hobby projects. I recently purchased three Arduino Nanos for only $5 each, which still blows me away. The ATmega328P microcontroller chip on one of these boards boasts a 16MHz clock, 32kiB of flash memory, 2kiB of SRAM and a 10-bit, 6-input ADC.


That's not an Arduino… **this** is an Arduino!

In 1975, 16MHz would have sounded like lab-grade or military-grade hardware; 32kiB of ROM would have required an entire board full of chips (and would have cost some serious money); and having a 10-bit ADC integrated into the process would have felt like something from NASA.

When I was a student in the late 1970s, even the humblest single-board computer (SBC) was far outside my means. I'm almost embarrassed to say that, while rooting through some boxes of bits at home the other day on the hunt for a piezoelectric buzzer (for use in my next Weird and Wonderful column), I must have run across at least five Arduino Uno boards lurking in various states of benign neglect.

In 1975, just one of these boards would have had engineers flocking around in wide-eyed disbelief!

Do you remember when the boy in *The Sixth Sense* says, "I see dead people"? I mistakenly remembered this as "I see dead people everywhere". The reason I mention this is that I'm starting to see Arduinos everywhere. As I pen these words, I'm sitting at the desk in my office at work, and I see at least three Arduino Uno boards peeking out at me from under piles of bits and pieces.

Furthermore, there's a small cardboard box sitting on a shelf in front of me that has been niggling me for ages because I couldn't recall what it contained, but I was too lazy to get up and see. I finally forced myself to look inside, only to discover a "Make Your Own Arduino" kit (**https://docs.arduino.cc/hardware/make-your-uno-kit/**).

Where on Earth did that come from? And what are the odds that it would surface while I was writing this column?

Speaking of DIY Arduinos, have you seen the giant version created by a guy who goes by the name of Uncle Stem? This really is a beauty to behold. He talks us through the build in a YouTube video (**https://youtu.be/RZkv9WKkpGg**).

This isn't just for show. It actually works, although paradoxically and/or ironically, it's powered by an Arduino Nano. You can upload programs to it and run them. The oversized onboard light-emitting diodes (LEDs) light up, and you can plug wires and external components into the header pins.

Of course, Uncle Stem has also created a set of large components, such as resistors and LEDs, that are to scale. Seeing this leads me to think of two things. First, Uncle Stem has too much time on his hands. And second, **I want one**!

## Feeling tensor?

It's often said that you don't realise what you don't know until you try to explain something you think you do know to someone else. I was reminded of this recently while listening to a friend who was attempting to explain the differences between central processing units (CPUs), digital signal processors (DSPs), graphical processing units (GPUs), and neural processing units (NPUs).

Of course, it's easy to be a critic (spotting other people's shortcomings is one of the things I do best), but it struck me that my friend might have been better served if he'd begun by laying a little groundwork by introducing the different types of data that need to be processed.

For example, in computer science, a scalar is a single value; a vector is a one-dimensional array of scalars; a matrix is a two-dimensional grid of scalars (which can also be viewed as a table of scalars, or as a collection of row or column vectors); and a tensor extends this to three or more dimensions (think of it as a stack of matrices, or a higher-dimensional stack).

These terms are used somewhat differently in other disciplines such as mathematics and physics (we had to overload their meanings, didn't we?).

You might find it helps to visualise this as a progression as follows:

- 0D: Scalar → one number
- 1D: Vector → a line of numbers
- 2D: Matrix → a table of numbers
- 3D (and higher): Tensor → a stack of tables (and possibly stacks of stacks). Beyond that, it gets a bit complicated.

*continued on page 75…*

Power Electronics series

By Andrew Levido

# Power Electronics

## Part 2: Controlling DC-DC Converters

**Last month, we introduced average value analysis and used it to analyse a range of DC-DC converter topologies at a high level. We then looked at a practical example of a buck converter. This time, we will delve into the control systems required to make these converters work correctly.**

**DC-DC converters almost always operate in a closed-loop mode,** where some kind of feedback is used to maintain the output voltage at the desired level, despite changes in load or source voltage. Often, the detail of the control loop is hidden from the user because it is incorporated in the driver chip, such as was the case for our buck converter example last month.

Even when this is the case, it is handy to have at least a basic understanding of how they work.

We will continue using the example of the buck converter we introduced last time. It was designed to convert a nominal 12V input (ranging from 10V to 14V) to a 6V output, to drive a 10Ω load for a maximum output current of 600mA. The switching frequency was 500kHz. We went through the process of selecting the filter components in some detail.

If you want to know why we need to close the loop around our converter, take a look at Fig.1. This is the result of a simulation of the converter with no feedback and a fixed 50% duty cycle.

The source voltage (blue trace) is programmed to be 12V for the first 30ms, dropping to 10V thereafter. In contrast, the load is programmed to be 10Ω for the first 10ms, increasing to 20Ω thereafter, with the resulting load current shown by the orange trace. The purple trace shows the output voltage.

There is a significant transient when the load changes, but no change in output voltage, since the duty cycle is fixed. The transient peaks at about ±300mV, which is not great, and it takes about 5ms to settle.

Things get even worse when the input voltage reduces at the 30ms mark. Not only does the output voltage drop because there is no regulation, but the ringing at the transition is horrendous. Again, it takes about 5ms for the result to settle into something close to the steady state value.

A properly designed closed-loop control system should greatly reduce or eliminate these problems.

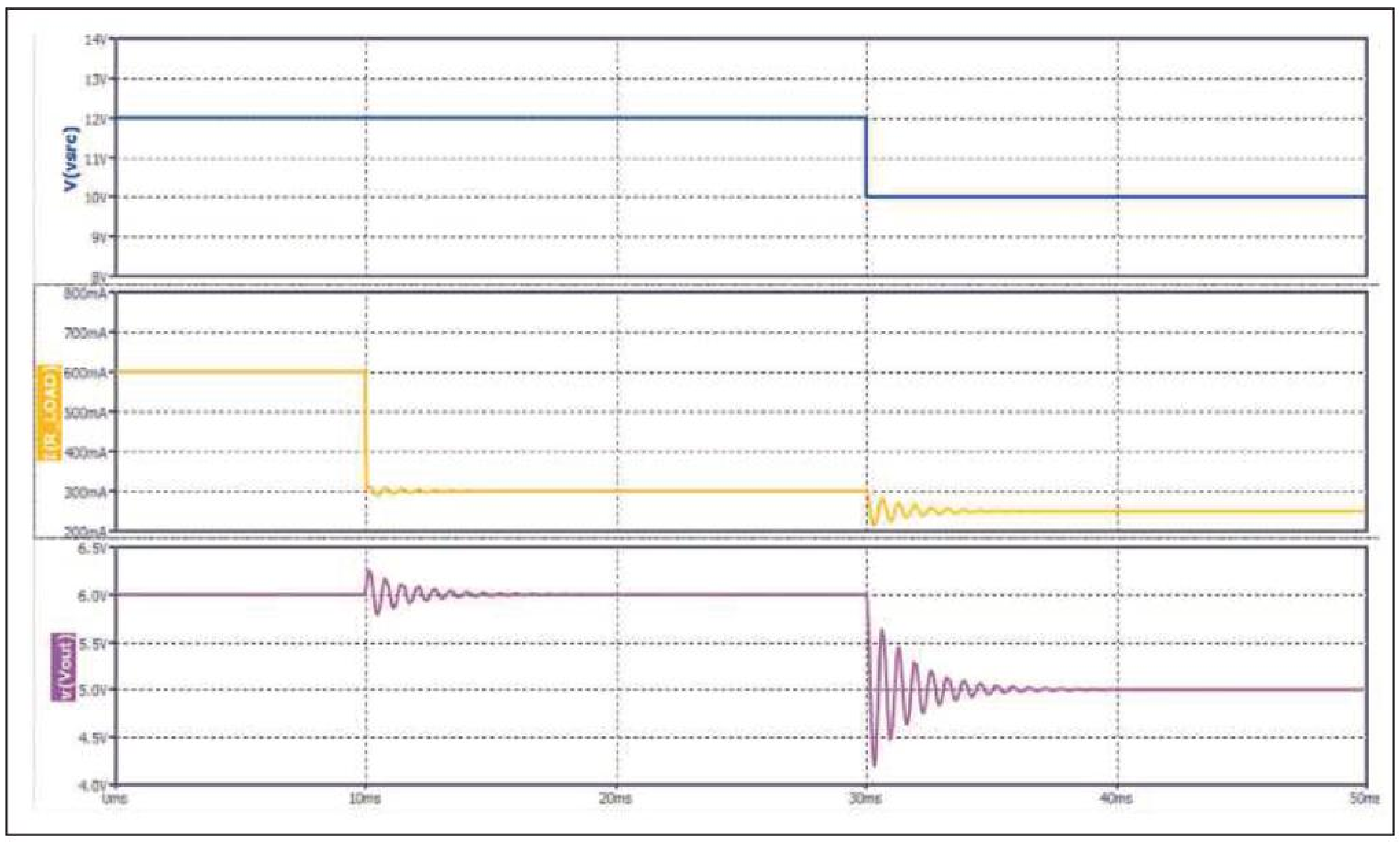


**Fig.1: without a closed loop control system, the output of the DC-DC converter (purple trace) is unstable with changes in load (at 10ms) and input voltage (at 30ms).**

Control theory is a huge topic, and one that can be very heavy on mathematics. I will therefore not be giving a control theory tutorial in the classical sense, but rather a practical exploration of some of the techniques that can be employed. If you are a control theory expert, you are hereby warned that I am going to skip some details in the interest of space and simplicity!

### Creating a mathematical model

To create a stable and effective control system, we have to build a mathematical model to describe our DC-DC converter under dynamic conditions. We have already discussed two models for our converter: the very high-level periodic steady state (PSS) model we used for average value analysis, and the very low-level cycle-by-cycle simulation model we used to verify the ripple voltages and currents.

We can't use the periodic steady-state model for the control loop since it is dynamic behaviour we want to take care of. Also, we don't need to model the cycle-by-cycle behaviour of the converter since the behaviour we are concerned with occurs over much longer time periods than a single 2µs cycle. Instead, we need an intermediate model.

Fig.2 shows the block diagram of our converter with feedback presented in the classic control theory manner. Right in the middle is the modulator, which takes a control voltage proportional to the duty cycle ($v_c$) and produces the switching waveform, whose average value, $\langle v_x \rangle$, is applied to the filter.

The filter takes this voltage and produces the smoothed output voltage, $v_o$. Each block has a gain, $G$, which relates its output signal to its input.

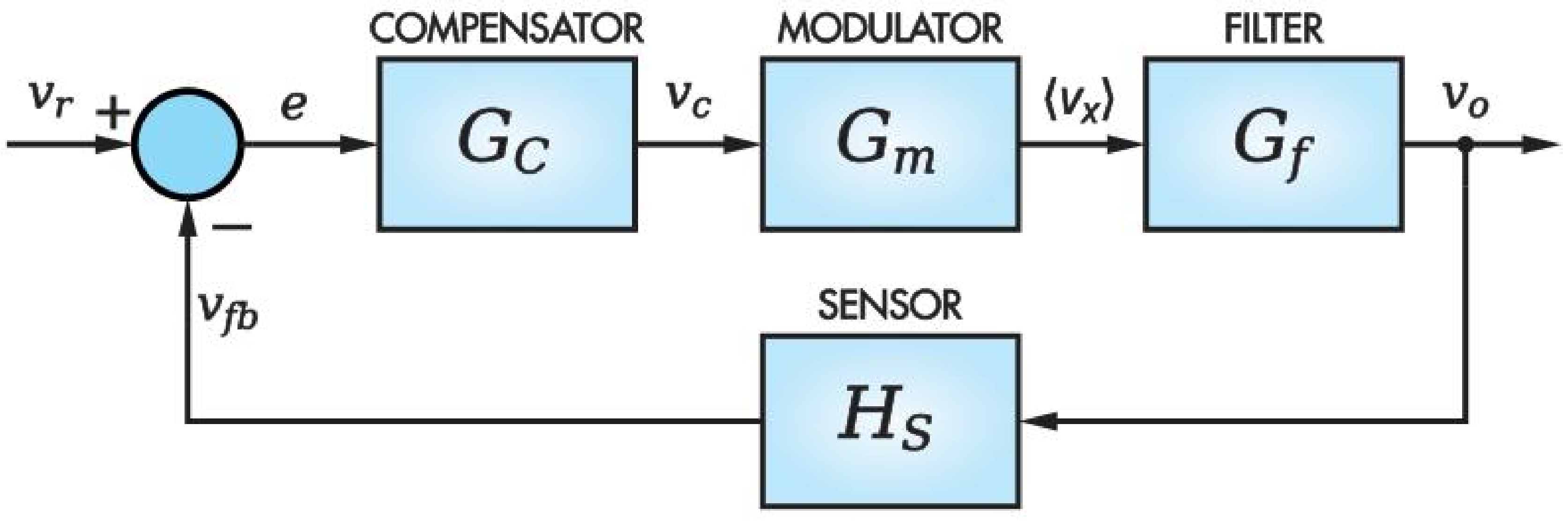


**Fig.2: this classic depiction of the DC-DC converter closed-loop system is useful to understand the circuit blocks we need to model.**

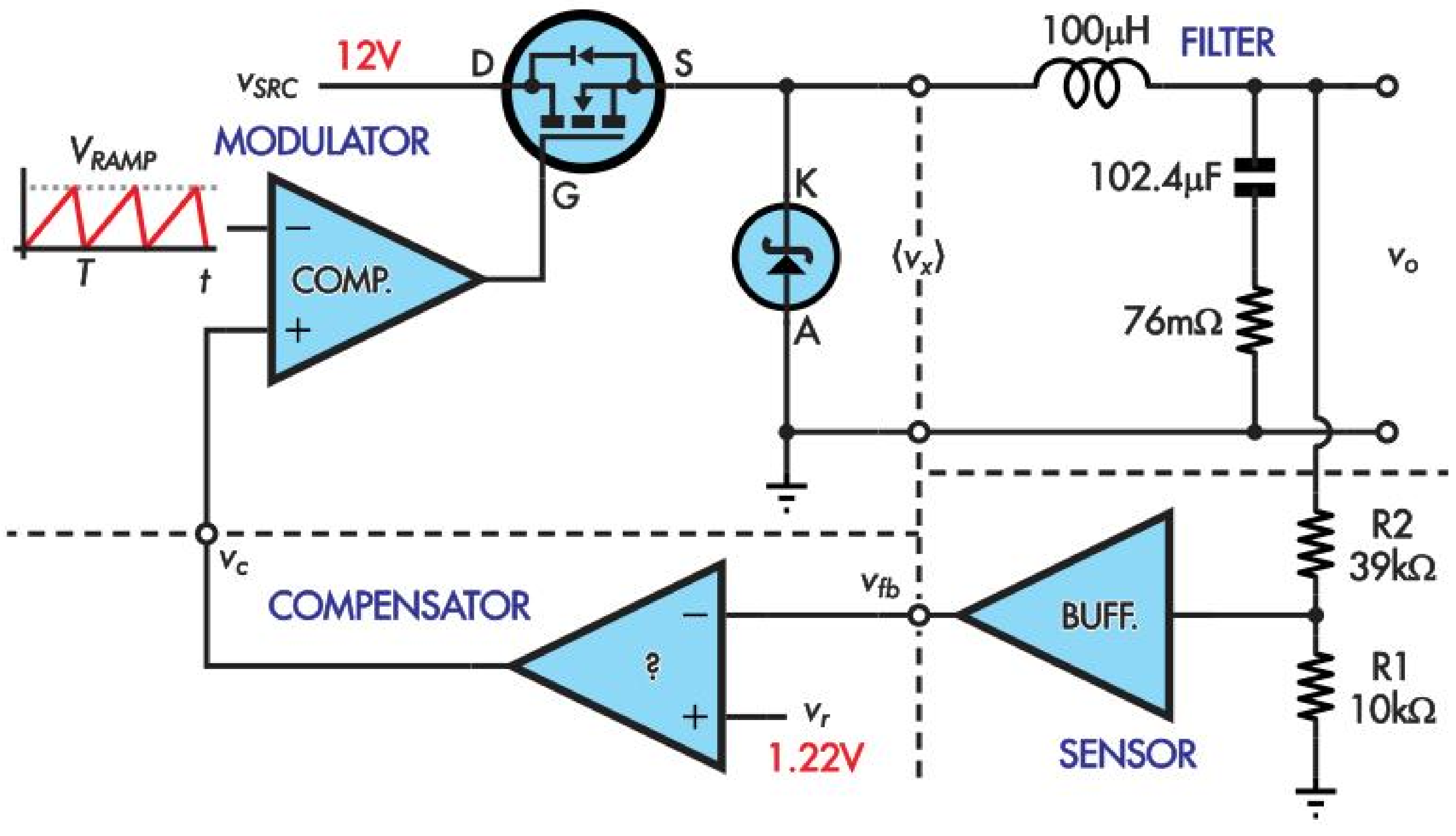


**Fig.3: the DC-DC converter arranged according to the block diagram in Fig.2 is the starting point for the development of a control model of the converter. The compensator is the missing piece of the puzzle at this stage.**

In front of these two blocks is a compensator, which we will describe in more detail below. The input to the compensator is an error, e, that is the difference between the reference setpoint $v_r$ and the feedback signal $v_{fb}$. The feedback signal is derived from the output voltage we want to control by some sensor, with a gain of $H$.

These blocks are shown in a more familiar way in Fig.3. The modulator includes the Mosfet and diode switches, plus the circuitry that converts a control voltage to pulse-width-modulation to drive the Mosfet. At its simplest, this is just a comparator with an output that is high whenever the control voltage exceeds a ramp waveform at the switching frequency.

The modulator therefore takes a control voltage, $v_C$, and produces the average voltage, $\langle v_x \rangle$. The modulator has a gain of $G_m$ and it is easy to see that this must be the unitless quantity $G_m = V_{SRC}/V_{RAMP}$. If we assume the ramp voltage is 1V, the modulator gain simplifies to the source voltage, which is 12 in our example.

It is worth noting that there is no need for any switching to occur in this model of the modulator. The modulator produces an output equal to the average voltage over one switching period. This voltage may vary with time, but the time scale we are concerned about is much longer than the 2µs switching period.

Because this modulator controls the average voltage applied to the filter, this converter is said to be employing voltage mode control. There is another way to do it, which we will come to later.

The sensor gain is trivial to calculate. It is simply the divider ratio represented by the two resistors, so $H_S$ = R1 ÷ (R1 + R2), or 0.204 in this case. The buffer is there to isolate the compensator from the divider.

We will ignore the compensator for the moment, but notice that I have absorbed the error summing junction into the black box and added a positive input for the reference and a negative input for the feedback. This anticipates that the compensator will probably involve op amps.

## Simplifying the complex-frequency domain

Describing the filter and compensator gains requires us to dip our toes into something called the complex-frequency domain, also known as the s-domain. Analysis of the behaviour of capacitors and inductors involves differential equations, which we usually describe in the time domain – for example, the relationship between voltage and current in an inductor is $v = L\, di/dt$.

We can describe the same equations in the s-domain, which makes the mathematics much simpler, because differentiation is reduced to multiplication and integration is reduced to division. The conversion from the time domain to the s-domain requires something called a Laplace transform.

You don't need to know how to do this (unless you are an electrical engineering student) because you can write the s-domain expression for most common components very easily without having to do any maths.

**Table 1 – passive component complex impedances**

| Component | Time Domain | s-Domain | Complex Impedance Z(s) |
|---|---|---|---|
| **Inductor** | v = L di/dt | v = iLs | Ls |
| **Capacitor** | i = C dv/dt | i = vCs | 1/Cs |
| **Resistor** | v = Ri | v = iR | R |

The "s" in s-domain is a complex frequency, which is just a way of describing a unit-amplitude sinewave with a given frequency and phase. We won't, however, have to deal with any complex numbers in this explanation – we can simply treat s like any other algebraic variable.

To transform the inductor mentioned above to the s-domain, the time differential (*d/dt*) is simply replaced by s to get the s-domain equivalent relationship $v = i \bullet L \bullet s$, which I hope you can see is kind of analogous to Ohm's law. The complex impedance of the inductor, indicated by *Z(s)*, is therefore simply Ls.

Using a similar logic, we can arrive at the s-domain expressions for the voltage/current relationships of common parts, and hence their complex impedances, shown in Table 1.

To show just how useful this transformation is, we will use these complex impedances to calculate the transfer function of a simple RC low-pass filter shown at the top of Fig.4. If you substitute in the complex impedances from the table and use the normal voltage divider equation, you get the s-domain transfer function 1 ÷ (RCs + 1) after a little bit of algebraic manipulation. It's that easy.

## Poles and zeros

You can see that there must be a particular value of complex frequency s that makes the denominator of the expression equal zero, and therefore the transfer function becomes undefined (but very large). This value, s = -1 ÷ (RC), is called a "pole" because you can visualise it as a spike rising to infinity on the two-dimensional s-plane.

The complex frequency domain uses radians per second as the unit of frequency, but converting to Hertz (via the relationship that one cycle is 2π radians), this pole is at a frequency of 1 ÷ (2πRC)Hz. This should be familiar as the corner frequency of an RC filter.

If we turn to the RC high-pass filter at the bottom of Fig.4 and go through the same process, we arrive at the s-domain transfer function RCs ÷ (RCs + 1). This too has a single pole, at –1 ÷ RC, but the s in the numerator means there is also a value of s where the numerator of the fraction becomes zero, so the transfer function is equal to zero.

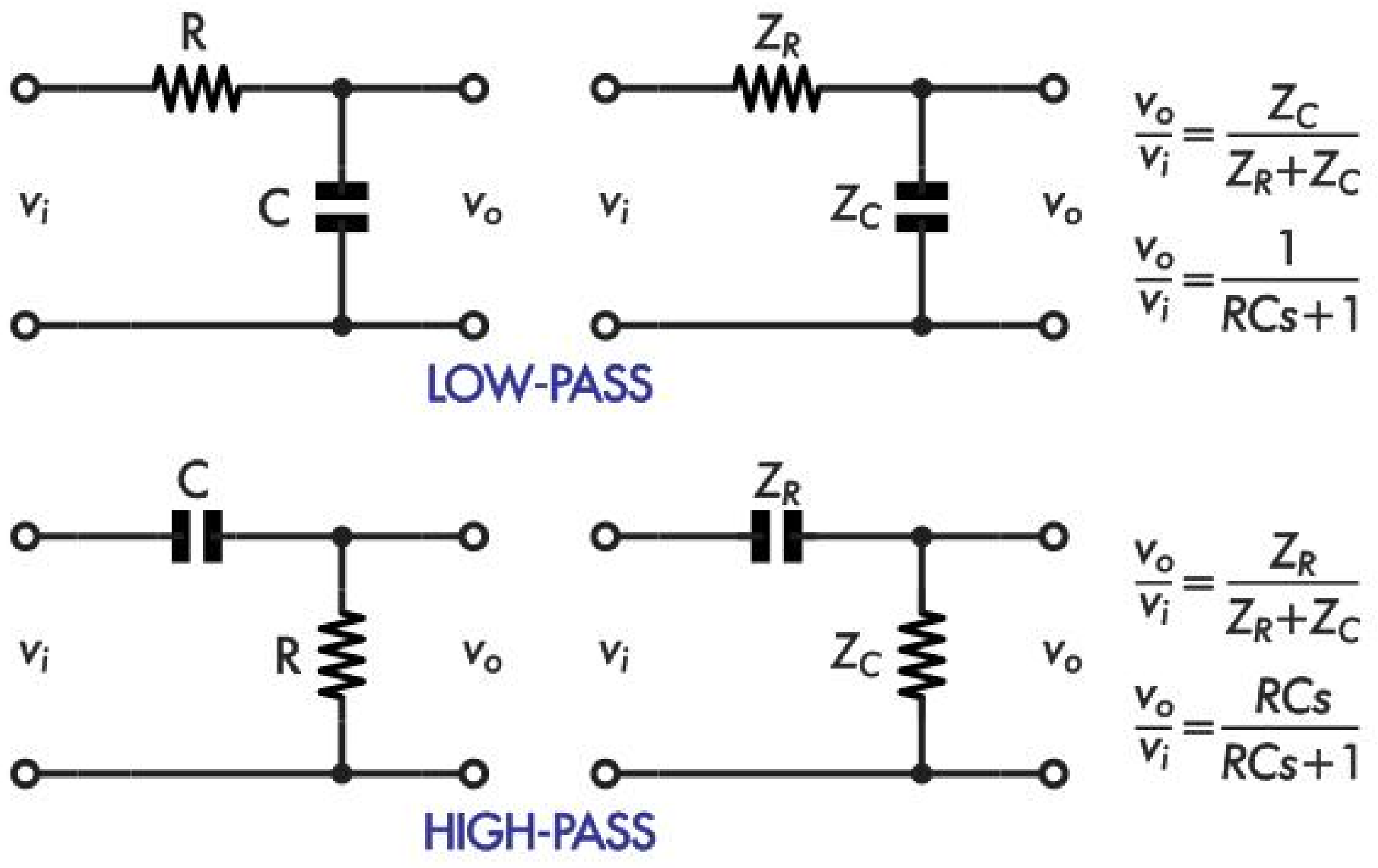
It may come as no surprise to know that this is called a "zero". In this case, the zero occurs at s = 0, so the LC high-pass filter has a transfer function with one pole at 1 ÷ (2πRC) and one zero at the origin. By origin, we really mean the origin of the s-plane, but you can think of this zero as being at "0Hz".

So, you can easily work out the poles & zeroes of any network by calculating the transfer function in the s-domain as a fraction. Poles are the values of s (frequency) when the denominator is zero, and zeroes are the values of s when the numerator is zero.

Poles & zeros have a specific meaning in the frequency domain. When a signal encounters a pole, the slope of the gain reduces by –20dB/decade and the phase shifts by –90°. Conversely, when a signal encounters a zero, the slope of its gain increases by 20dB/decade and the phase shifts by +90°.

Fig.5 shows this in action for the two filters we just analysed. In the case of the low-pass filter, the signal is unattenuated and has a phase shift of zero until it encounters the pole at which point it turns down to a slope of –20dB per decade. At the same time, the phase shifts by –90°.

**Fig.4: finding the s-domain transfer functions for these filters is as simple as substituting the expressions from Table 1 into the voltage divider equation and performing some elementary algebra.**


R
C
$v_i$
$v_o$
$Z_R$
$Z_C$
$\frac{v_o}{v_i} = \frac{Z_C}{Z_R + Z_C}$
$\frac{v_o}{v_i} = \frac{1}{RCs + 1}$
LOW-PASS
C
R
$Z_R$
$Z_C$
$v_i$
$v_o$
$\frac{v_o}{v_i} = \frac{Z_R}{Z_R + Z_C}$
$\frac{v_o}{v_i} = \frac{RCs}{RCs + 1}$
HIGH-PASS

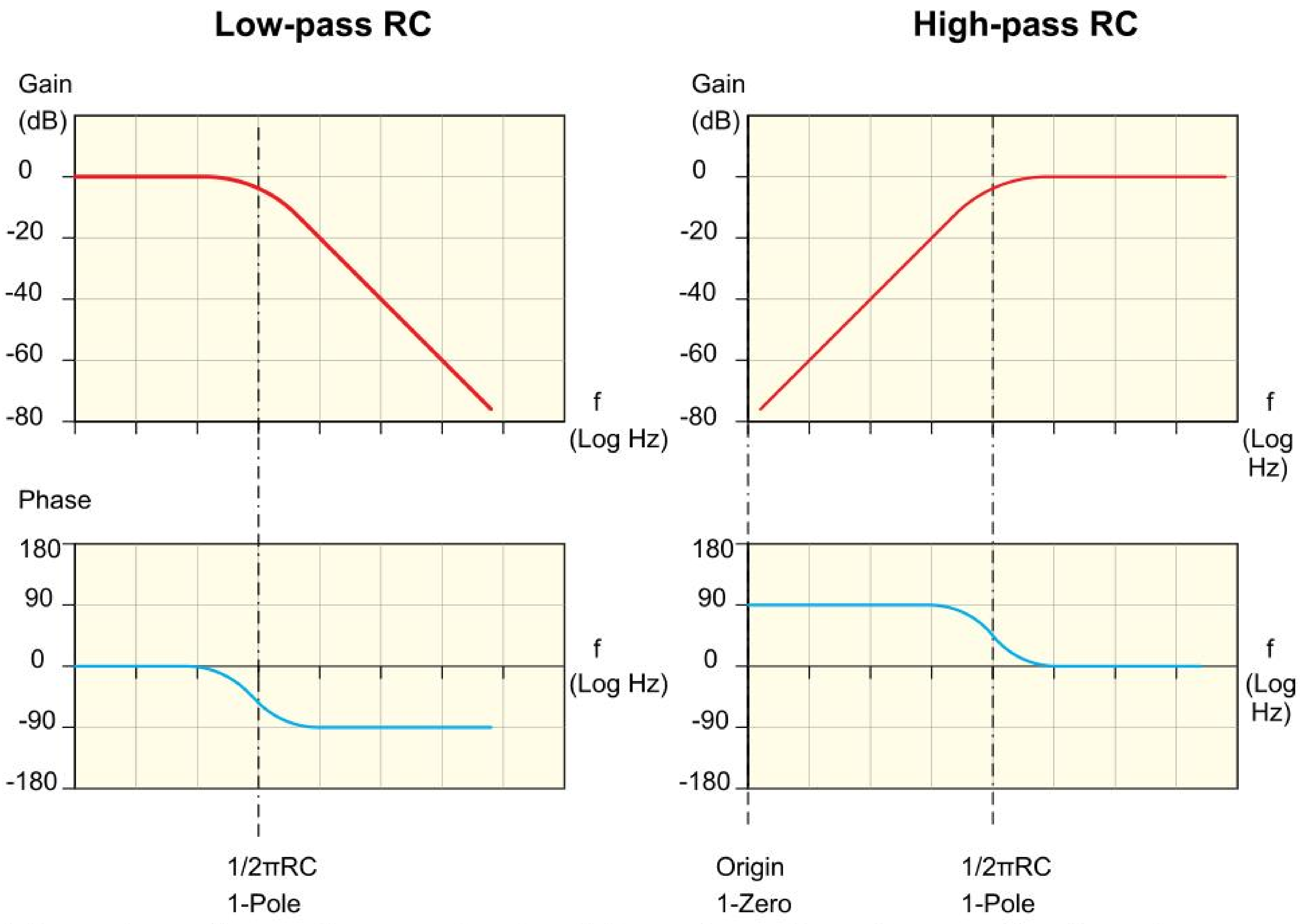


**Fig.5: the poles and zeros obtained from the s-domain transfer functions in Fig.4 can be easily transferred to the frequency response plots. A pole introduces a –20dB/decade slope change and a –90° phase shift. A zero introduces a +20dB/decade slope change and a +90° phase shift.**

In the case of the high-pass filter, the zero at the origin means that the gain is increasing by +20dB per decade from 'zero' and the phase is already shifted by +90°. When the pole is encountered, the –20dB/decade shift puts the gain back to flat, and the –90° shift brings the phase back to zero.

In practice, this means that you can work out the transfer function, the pole and zero locations and therefore the gain/phase versus frequency characteristic of any network of resistors, capacitors and inductors with a bit of high-school algebra. No wonder engineers like the s-domain. Thanks, Monsieur Laplace!

## Buck converter output filter

Armed with this new knowledge, we are ready to work out the transfer function, $G_f$, of the buck converter's output filter to fill out our control system model. This is shown on the left-hand side of Fig.6.

If you have a go at this yourself, you will see I have made a small approximation in the denominator, where an RCs term is dropped because it is very much smaller than the dominant $RLs^2$ term. This kind of simplification is common in complex frequency analysis, so watch out for them.

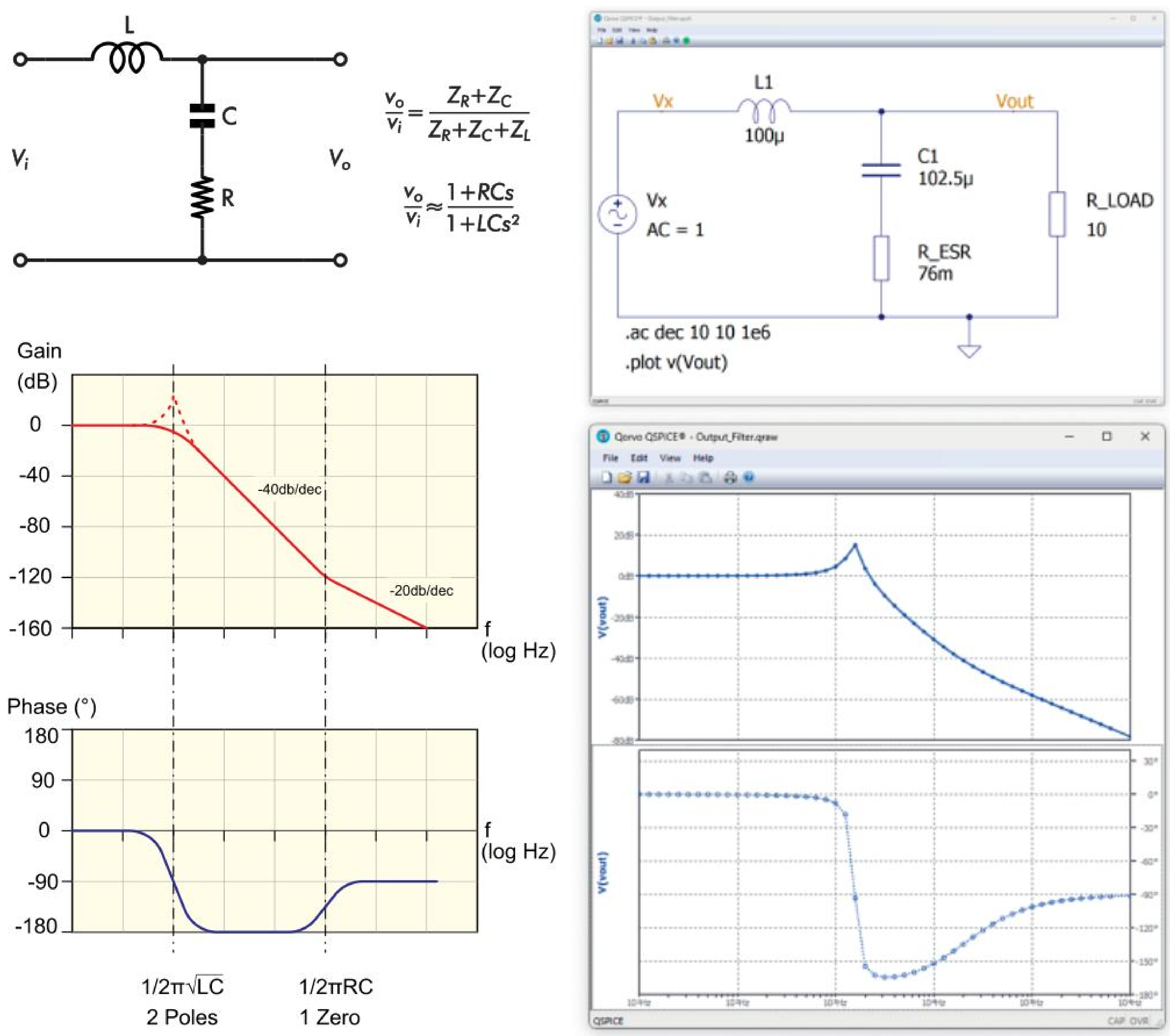
The resulting transfer function has an $s^2$ term in the denominator. This signifies that the transfer function has two poles; in this case, both are at $1 \div (2\pi\sqrt{LC})$. This means the roll-off at this frequency is –40dB/decade, and the phase shifts by –180°.

There is also one zero at 1 ÷ 2πRC due to the ESR of the of the capacitance. This pulls the gain back up to –20dB/decade and the phase back up to –90°.

This double-pole is a problem for our converter. Gain 'peaking', like that shown dotted due to the Q of the LC filter, and the rapid phase transition can cause instability like the ringing we saw in Fig.1. If we compare this plot to the simulation on the right-hand side of Fig.6, you can see that the peaking extends to about 15dB, which is potentially quite serious.

## Designing a compensator

Finally, we come to the design of the compensator. There are several ways to go about this, but they all come



```
.ac dec 10 10 1e6
.plot v(Vout)
```

**Fig.6: the DC-DC converter output filter has two poles at a frequency determined by the inductance and capacitance, and a zero at the frequency of the capacitance and its ESR. The resulting frequency response contains a nasty peak & steep phase shift responsible for the ringing in Fig.1.**

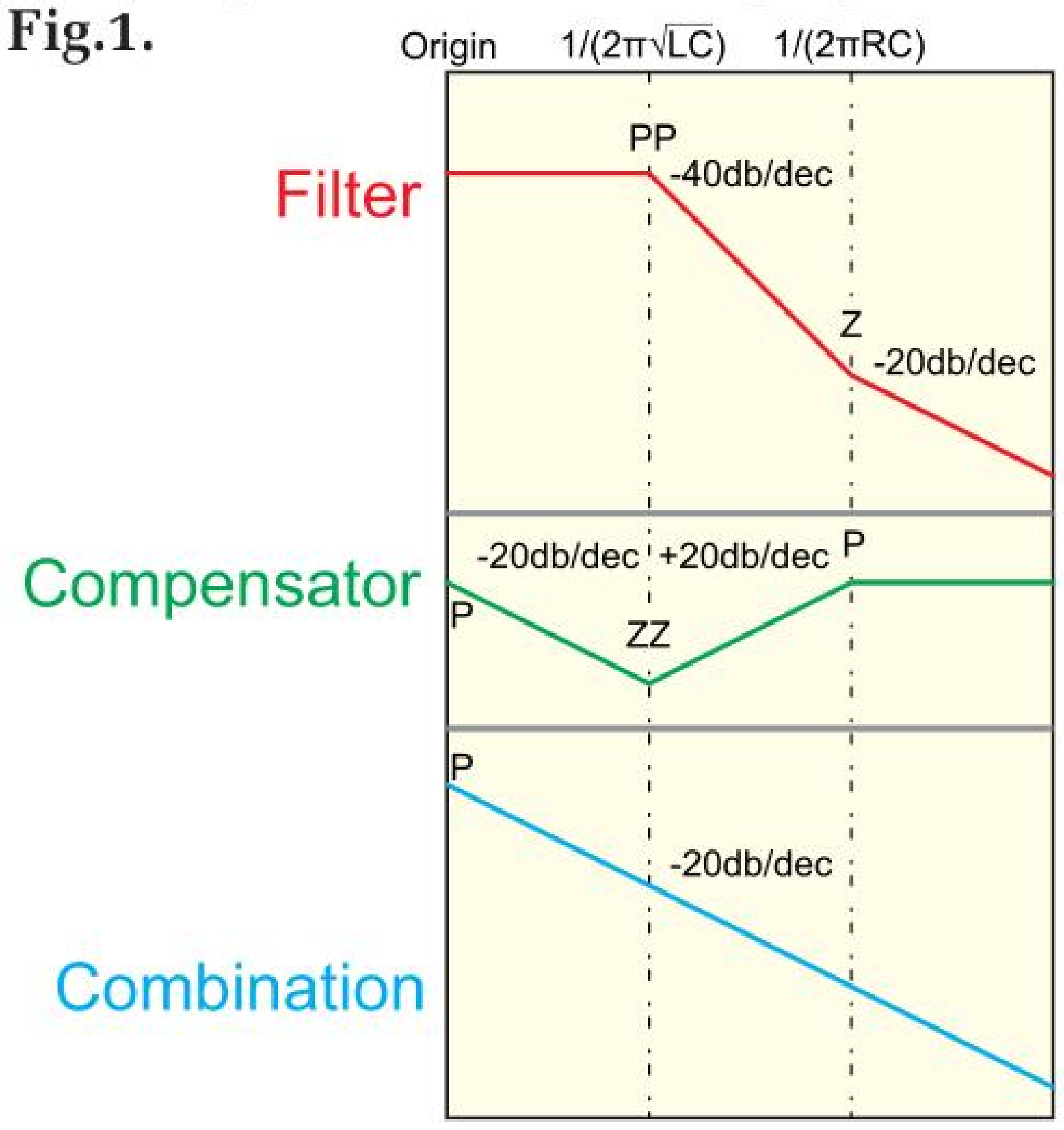


**Fig.7: we create a compensator characteristic (🟩) to transform the filter response into the dominant pole characteristic (🟦). The compensator requires two poles & two zeroes.**

down to manipulating the poles and zeros of the closed-loop transfer function. Their number, position and relationship can be tweaked to ensure the system is stable and to optimise the transient response.

For example, if we are most worried about overshoot, we can set up a highly damped system that will eliminate it at the expense of response time and settling time. On the other hand, we might be concerned about getting to the new voltage quickly, and can therefore tolerate a bit of overshoot.

I'm not going to dig into all of that here. Instead, I will demonstrate a simple technique called dominant-pole compensation that focuses on the open-loop poles and zeros. This is the same technique used to ensure op amps are stable in closed-loop applications.

The aim is to make the open-loop frequency response look like a single-pole filter that rolls off the gain at a steady –20dB/decade until it reaches 0dB at some crossover frequency, $f_C$. This roll-off is required to ensure the gain drops to unity well before the phase shift reaches –180°.

The difference between the phase shift at the crossover frequency and –180° is the 'phase margin', a measure of the stability of the circuit and its tolerance to external disturbances.

We can achieve this roll-off by adding a compensator with the characteristics shown in Fig.7. The modulator and the sensor gains are not frequency-dependent, so they don't impact the shape of the gain/frequency characteristic of the open-loop converter, and can be ignored. They do impact the absolute value of the gain, but we are not concerned with that here.

If we want to transform the filter gain/frequency characteristic (in red) into the dominant pole characteristic in blue, we require the compensator to have the green characteristic.

It should have a pole at the origin, to set the roll-off to –20dB/decade. We then need two zeroes at the corner frequency of the LC filter to cancel the double-pole in the filter response. The compensator should have another pole at the frequency of the output capacitance/ESR zero to cancel the corresponding zero in the filter.

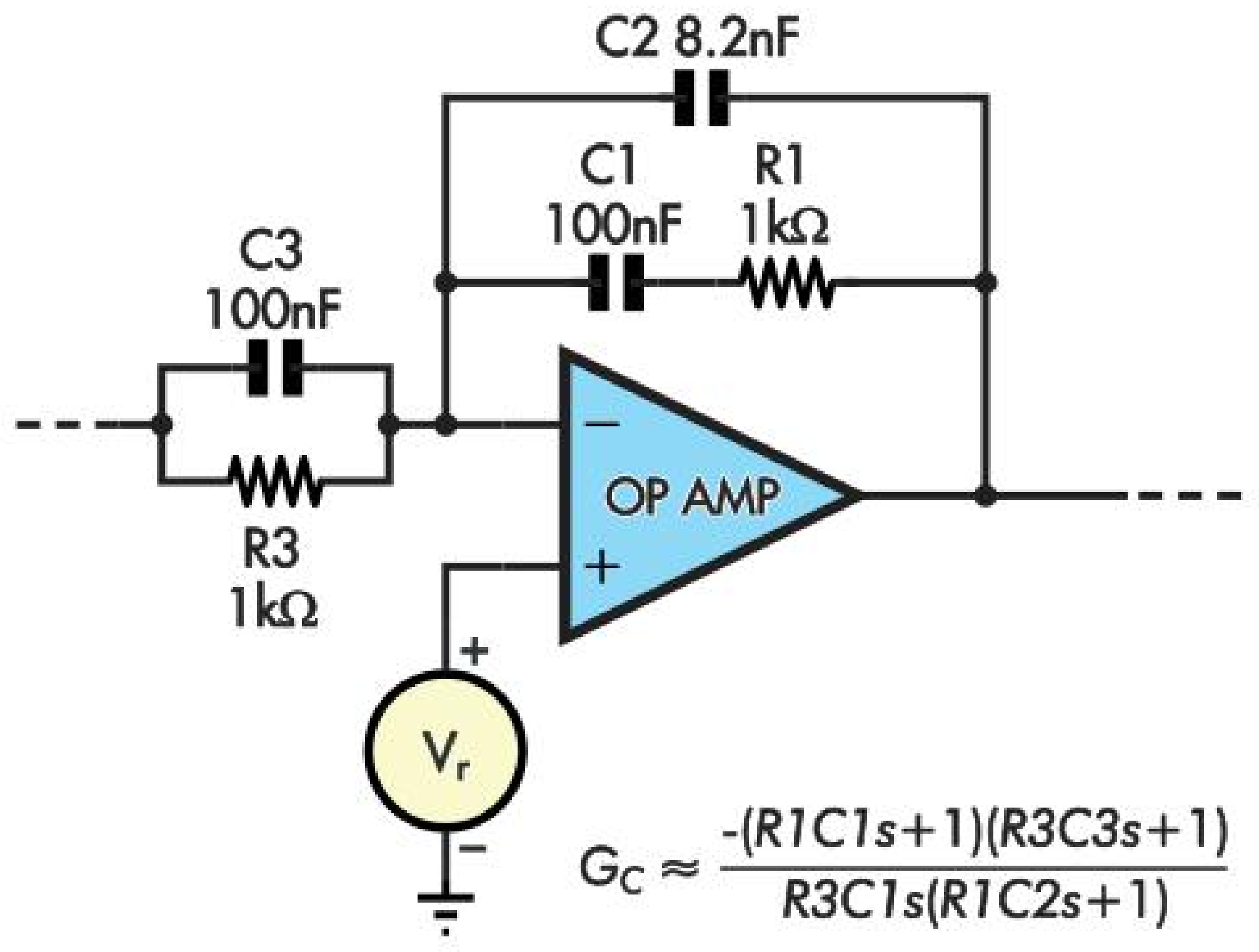


**Fig.8: this Type III compensator has the necessary poles and zeroes. The transfer function is calculated in exactly the same way as for the filter; the component values are discussed in the text.**

So in total, we need a compensator with two zeroes and two poles, one of which is at the origin.

This is known as a Type III compensator, and a suitable circuit is shown in Fig.8. The reference voltage is applied to the non-inverting input of the op amp, and the feedback signal is applied to the inverting input via R3/C3.

You can work out the compensator's transfer function in the s-domain in exactly the same way as we did for the filter. This time, we just assume that $v_r$ is zero, and use the usual equation for the gain of an inverting amplifier: $G = -Z_f / Z_i$, where $Z_f$ is the impedance of C2 in parallel with R1 plus C1, and $Z_i$ is the impedance of R3 in parallel with C3.

The resulting transfer function (again with a minor simplification) is shown below the circuit diagram. There are two zeros, one formed by R1C1 and one by R3C3, and two poles, one at the origin and one formed by R1C2.

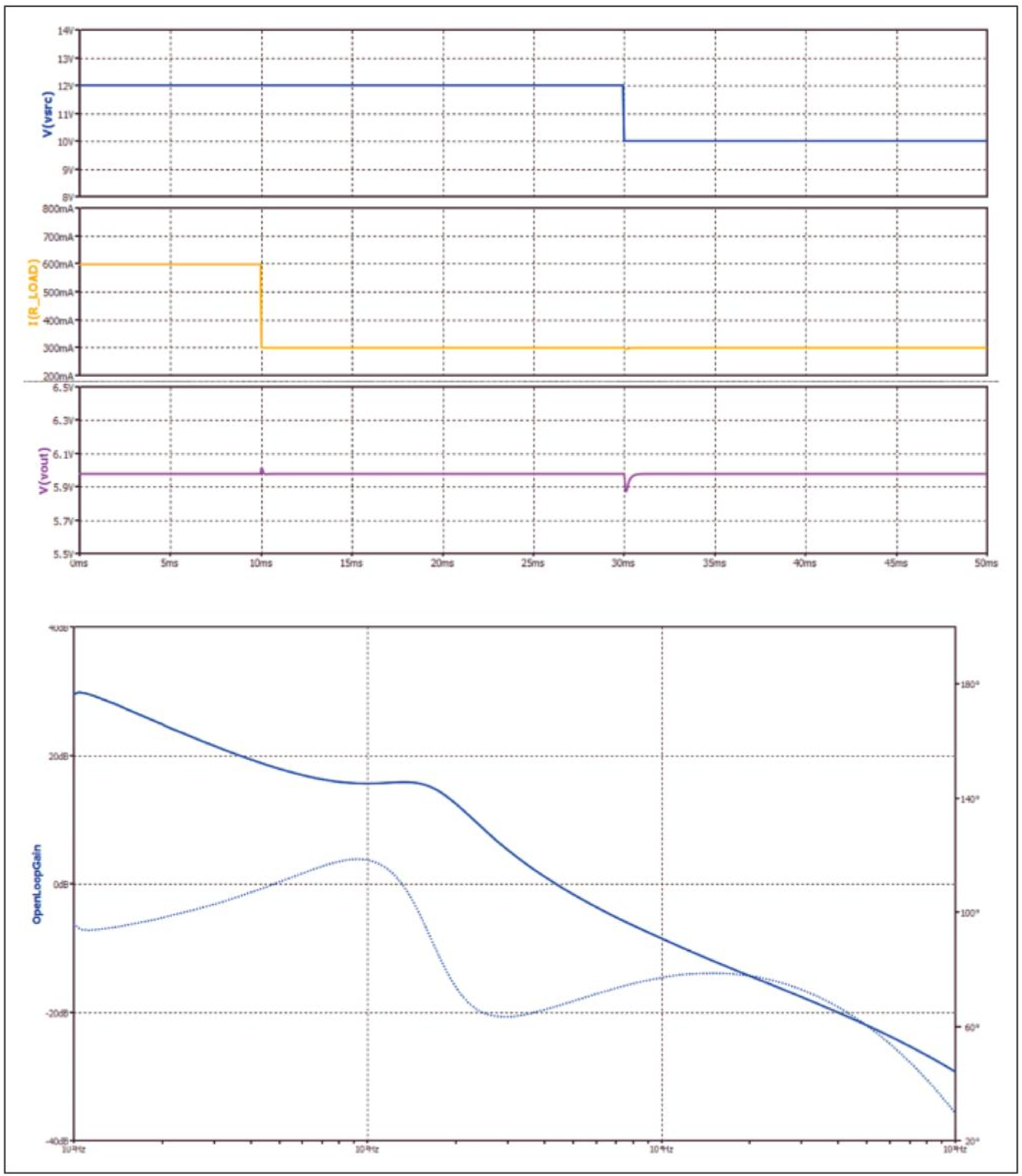


Fig.9: compare this closed-loop response to that of Fig.1. The disturbances when the load or supply voltage change are much smaller and settle much faster. The lower graph shows the open loop gain and phase with the compensator included. Contrast this with the filter characteristic in Fig.6.

The component values shown are calculated by setting the frequency of both zeroes, 1 ÷ (2πR1C1) and 1 ÷ (2πR3C3), to be the same as the LC filter's 1 ÷ (2π$\sqrt{LC}$), which is 1.56kHz. I chose 100nF and 1kΩ for nice, round values.

The non-origin pole at 1 ÷ (2πR1C2) should be at the same frequency as the zero formed by the output capacitor and its ESR (20.4kHz). Since we have already chosen R1, we can calculate C2 to be 7.8nF. This is not a standard value, so I used 8.2nF, the nearest one.

This small error won't make a huge difference, since the values for output capacitance and ESR we are using (102.4μF and 76mΩ) are hardly precision values themselves.

## Simulation

I simulated this control circuit to see how it will perform. The result is shown at the top of Fig.9. The stimulus is exactly the same as for Fig.1 – the load resistance changes from 10Ω to 20Ω at the 10ms mark, and the source voltage drops from 12V to 10V at the 30ms mark.

The difference is dramatic. There is just a small blip at the 10ms mark, which peaks at about +20mV (+0.3%), and lasts for just 150μs. At the 30ms mark, the voltage dips about 130mV when the input changes, but it recovers without overshoot in around 600μs.

I ran a second simulation to plot the open-loop gain of the entire circuit, which is shown at the bottom of Fig.9. While you can clearly see the expected -20dB/decade roll-off from 100Hz to 100kHz, a small amount of gain peaking is still visible, but it is much less pronounced. The simulation also calculates the crossover frequency, which is 4.36kHz, and the phase margin, which is 66°.

You may recall that the crossover frequency we calculated from the TPS5410 data sheet was 4.5kHz, so it seems likely the chip uses a similar compensator internally.

## Current-mode control

We mentioned above that there is a second type of modulator that can be used in DC-DC converters. This is known as current-mode control, and it is shown in Fig.10. In this case, the modulator controls the average current through the inductor, $\langle i_L \rangle$, instead of the average voltage at the filter input.

At the beginning of every switching period T, the latch is set, so the Mosfet switches on and the inductor current begins to increase. This current is monitored by a current sensor, usually a resistor, and the resulting voltage is compared to the control voltage, $v_C$. When the current reaches the level defined by the control voltage, the latch is reset and the Mosfet switches off until the next cycle.

The result is that the peak current is determined by the control voltage and the value of the sense resistor. If the current ripple is small, this turns out to be a reasonable approximation to the average current. This approximation is not perfect, so most real-life implementations use something called slope compensation to avoid a potential instability at very high duty cycles.

This is not relevant for the discussion below, so I will not go into it here.

The transfer function of a current mode modulator is therefore approximated by $G_m = 1 \div R_{SENSE}$. Note that this gain is a transconductance (a voltage to current conversion).

Because the output of the modulator is now a current, we have to revisit the transfer function of the output filter and the compensator, since the filter is no longer the voltage divider we modelled earlier.

Fig.11 shows the calculation of the filter transfer function, which is performed in exactly the same way as before.

What is different is that the inductor is effectively irrelevant, since the modulator current passes through it (you can think of the inductor as part of the modulator if that helps).

# Pt. 2: Controlling DC-DC Converters

The output voltage is now dependent on the impedance of the filter capacitor, including its ESR, and the impedance of the load.

The upshot of this is that the transfer function is simpler, with a single pole at a frequency dependent on the output capacitance and the load resistance, and a zero at the frequency related to the capacitance and its ESR, like before.

It follows that our compensator can be simpler, as it now needs only two poles and one zero, as shown at the bottom of Fig.11. This is known as a Type II compensator, and an example is shown in Fig.12.

The component values of R1 & C1 are calculated so that the frequency of the zero (1 ÷ 2πR1C1) is the same as the zero formed by the output capacitor and the load, at $1 \div 2\pi R_{LOAD}C$, which is 156Hz.

The pole at 1 ÷ 2πR1C2 should be at the same frequency as the zero formed by the output capacitor and its ESR (20.4kHz). Given that R1 is 10kΩ, the closest standard capacitor value comes out to be 820pF. The value of R3 should be very low compared to R1, and can even be zero.

I simulated this converter as well, although in this case, a change in input voltage was not modelled. Fig.14 on page 74 shows the results with a much magnified scale. The peak excursion and recovery time are very similar to the voltage mode controller, as we would expect.

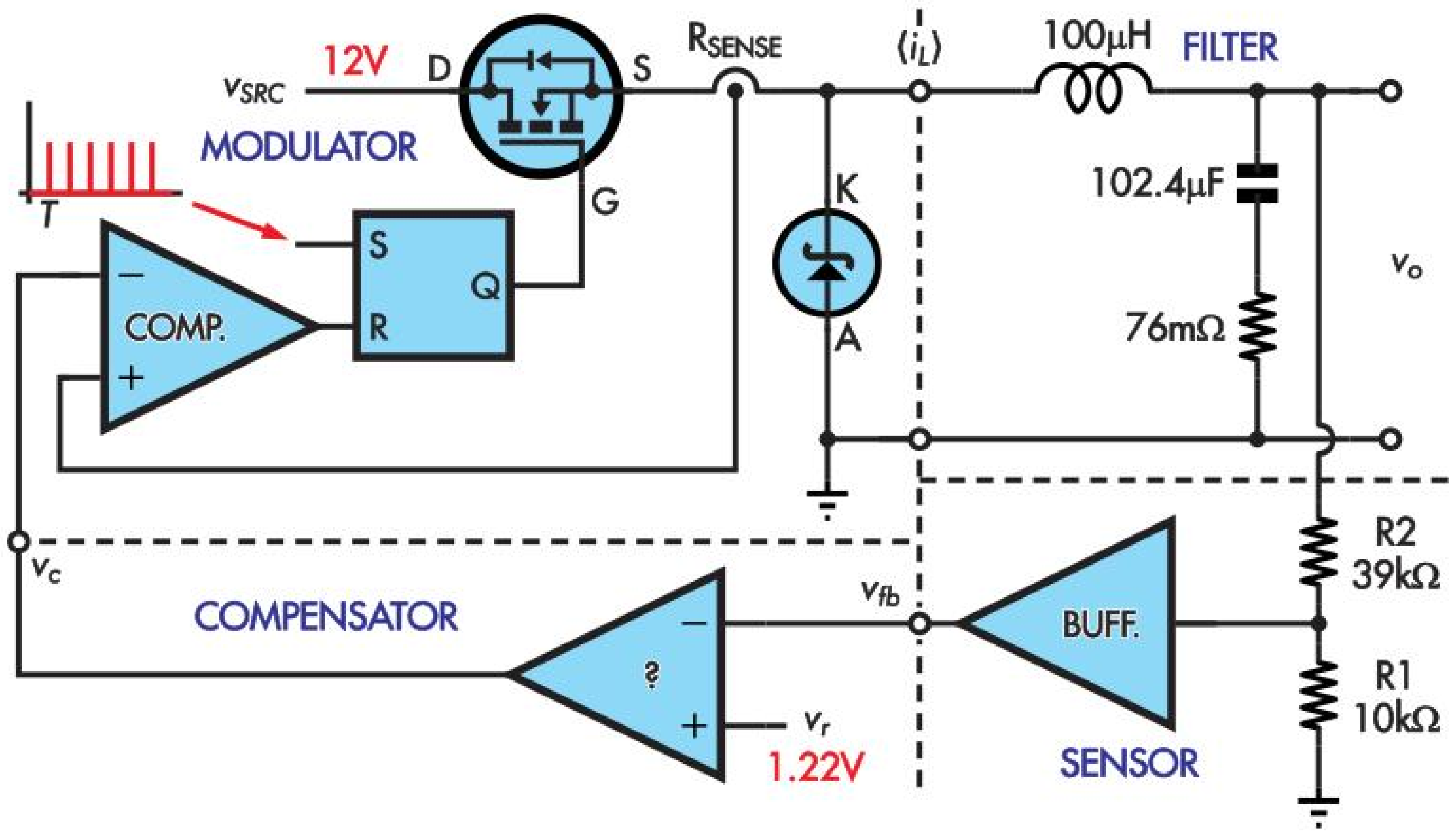
## Discontinuous current

Before we finish with control theory for now, we have to cover one more topic. So far, we have assumed that the current through the converter is continuous. That is, we have assumed the peak-to-peak current is small compared to the average. But what happens as the average inductor drops to the point it is close to the peak-to-peak ripple?

Fig.13 shows our old friend, the buck converter, where we have plotted the inductor current and the voltage $v_x$ that we have used in all of our analyses.

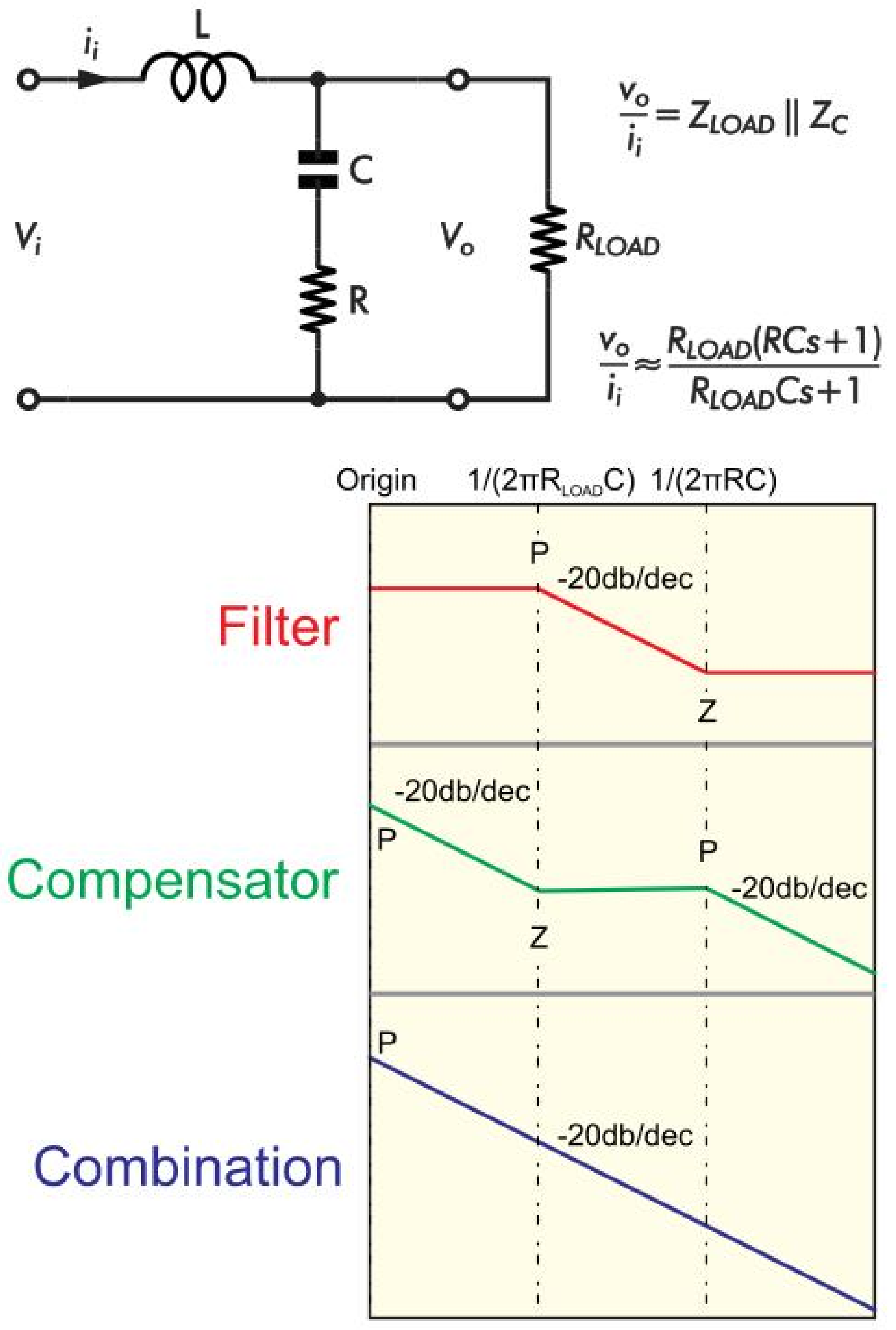
On the left is the case when the inductor current is high enough that the negative excursion of the (exaggerated) current ripple remains positive. This is known as the continuous current case, and has been the assumption we have used to derive

**Fig.10: a current-mode modulator controls the average inductor current instead of the average voltage as in Fig.3. This change flows on to the filter and compensator transfer functions.**

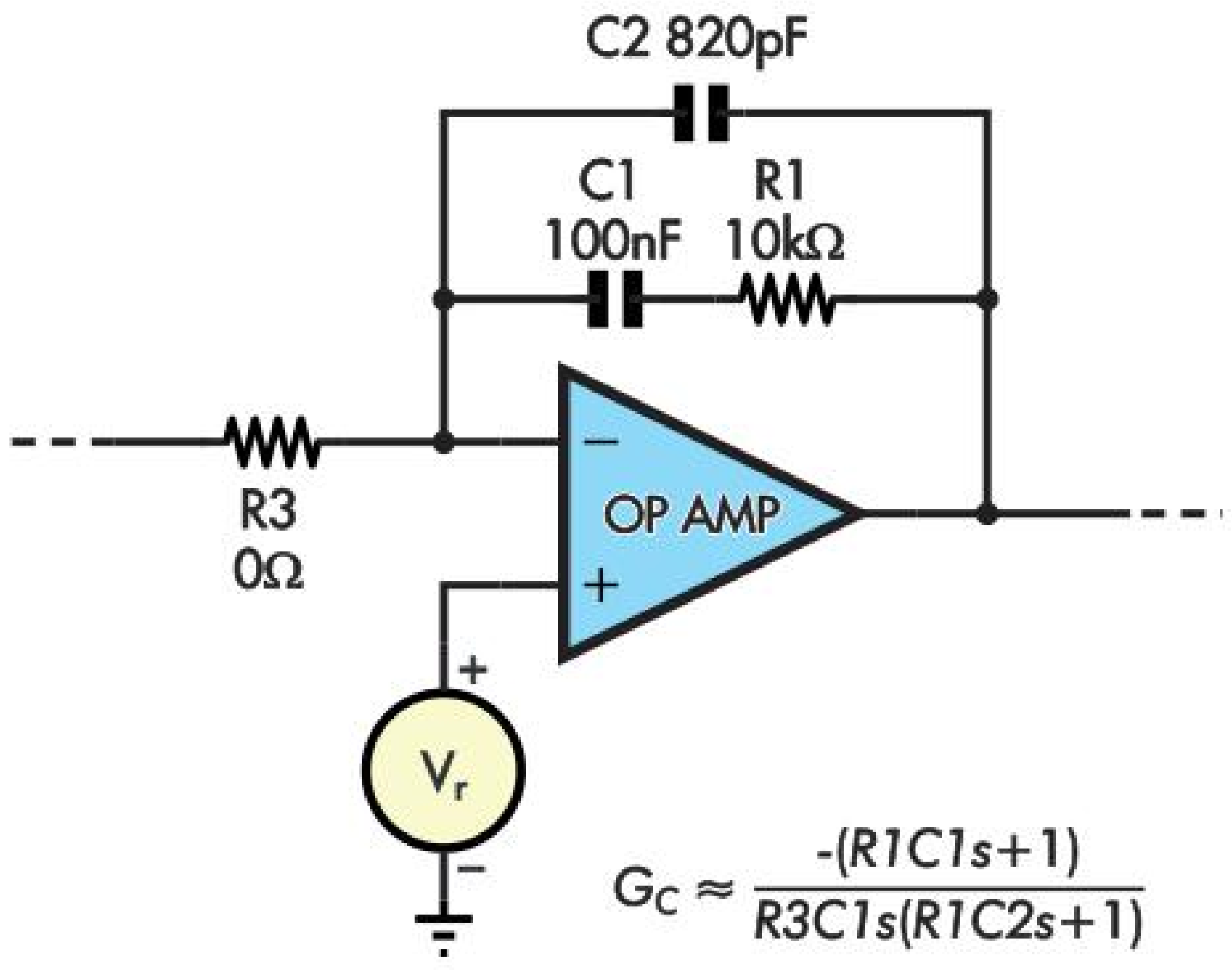


12V
vSRC
D
S
G
RSENSE
⟨iL⟩
100µH
FILTER
MODULATOR
T
S
R
Q
COMP.
K
A
102.4µF
76mΩ
vo
R2 39kΩ
R1 10kΩ
vc
COMPENSATOR
?
vfb
vr
1.22V
BUFF.
SENSOR

**Fig.11: the current-mode filter transfer function is simpler, with just one pole set by the capacitor and the load resistance and one zero due to the capacitor and its ESR. The compensator therefore only needs two poles and one zero.**

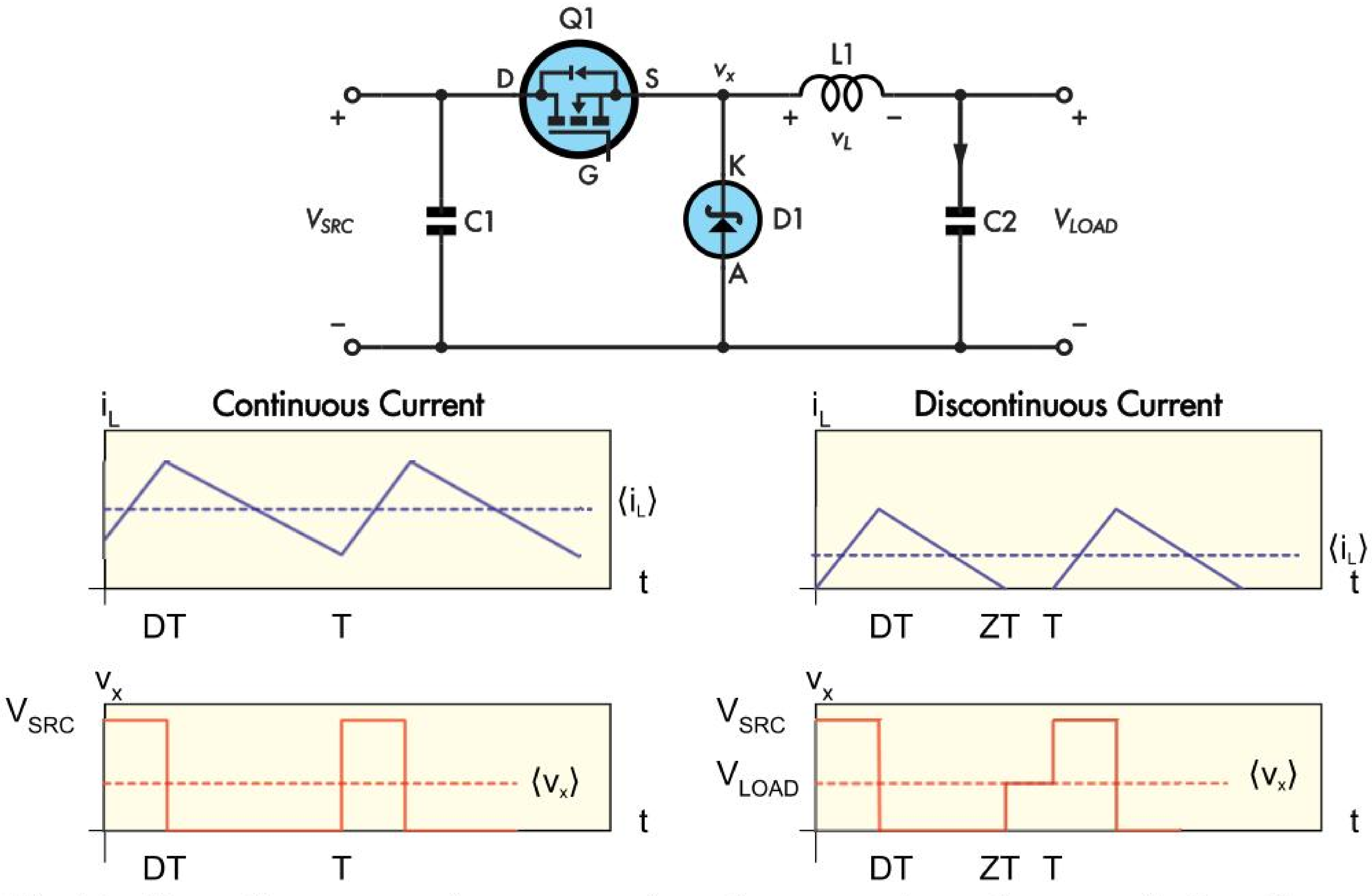


ii
L
C
R
Vi
Vo
RLOAD
$\frac{v_o}{i_i} = Z_{LOAD} \parallel Z_C$
$\frac{v_o}{i_i} \approx \frac{R_{LOAD}(RCs+1)}{R_{LOAD}Cs+1}$
Origin
1/(2πRLOADC)
1/(2πRC)
Filter
P
-20db/dec
Z
Compensator
-20db/dec
P
Z
P
-20db/dec
Combination
P
-20db/dec



C2 820pF
C1 100nF
R1 10kΩ
R3 0Ω
OP AMP
Vr
$G_C \approx \frac{-(R1C1s+1)}{R3C1s(R1C2s+1)}$

**Fig.12: this Type II compensator has the requisite poles (one at the origin) and one zero to compensate the current-mode controller. See the text for the component values.**



Q1
D
S
G
vx
L1
vL
VSRC
C1
K
D1
A
C2
VLOAD
Continuous Current
iL
⟨iL⟩
t
DT
T
Discontinuous Current
iL
⟨iL⟩
t
DT
ZT
T
vx
VSRC
⟨vx⟩
t
DT
T
vx
VSRC
VLOAD
⟨vx⟩
t
DT
ZT
T

**Fig.13: discontinuous mode occurs when the current reaches zero before the end of the switching period. This changes the modulator transfer function, and thus the compensator is required for optimum performance.**

the modulator models for both voltage mode and current mode examples above.

We could, however, imagine a situation when the load and therefore inductor current reduces to the point that the inductor current ramps down to zero at some time ($Z_T$) before the end of the switching period, as shown on the right. This is known as discontinuous current operation.

When the inductor current falls to zero, the inductor voltage must also be zero, so the point $v_x$ will be at the output voltage, $V_{LOAD}$.

It's pretty easy to see that the modulator transfer function will not be the same as for the continuous conduction case. The fact that the transfer function changes in discontinuous mode has implications for our control loop.

You can design an optimised control loop for a switching converter operating in either mode, but compromises are required if the converter will operate in both modes. Converters are therefore generally designed to work in one mode or the other. High-power converters usually use continuous current mode, and thus have a specified minimum output current.

Low power converters (say, ≤50W) often operate exclusively in discontinuous current mode. Of course, depending on what you are powering, it may be necessary to operate in both modes, in which case compromises much be made to ensure stability while keeping reasonably good load and line regulation.

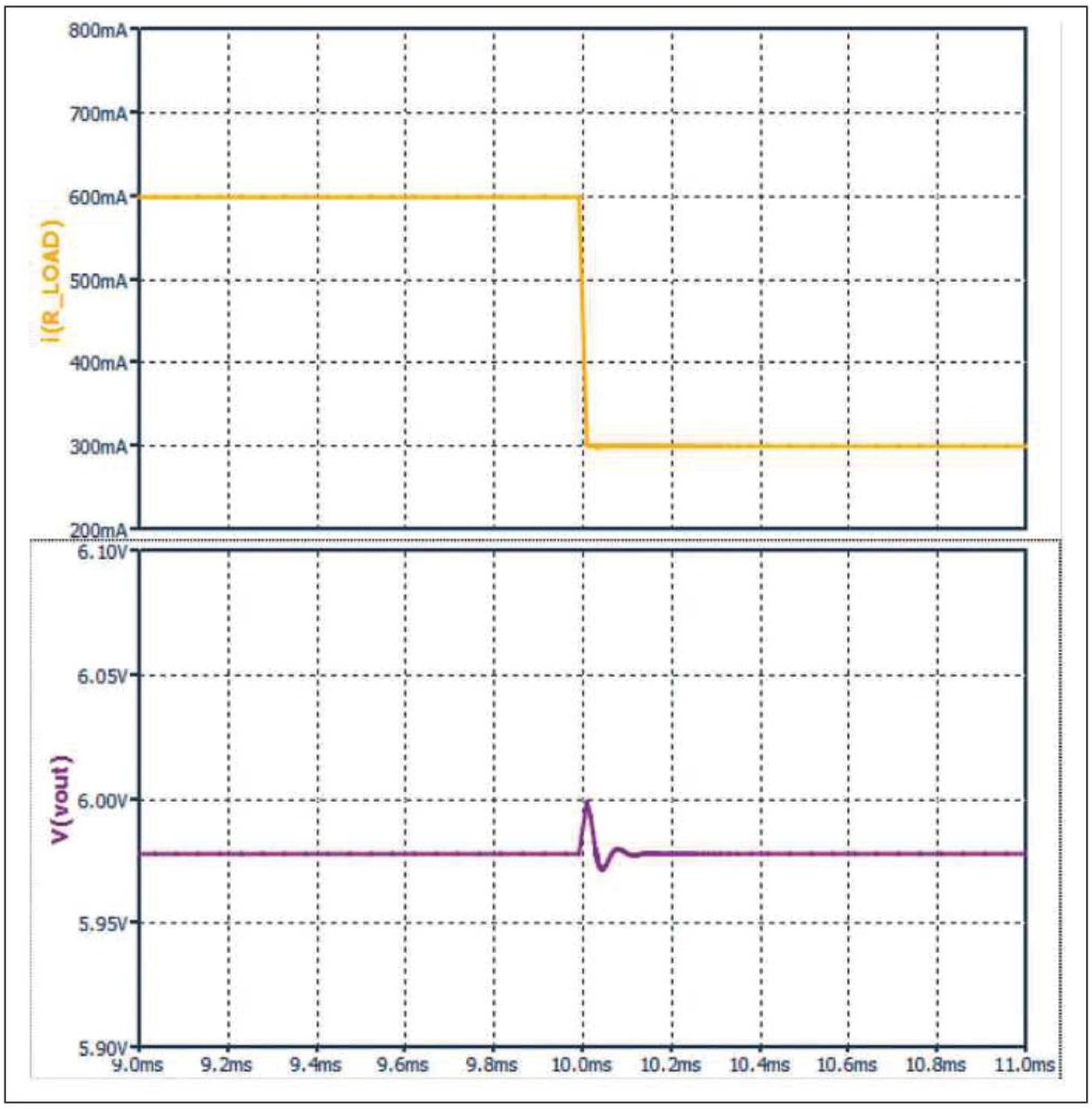


**Fig.14: the resulting closed-loop response to a load change is similar to that of the voltage mode controller (albeit with a different scale).**

## Conclusion

Whew! This has been a pretty heavy article, but I think it is important to have a basic understanding of the principles of control theory as applied to DC-DC converters.

You may never have to completely design a compensator from scratch, as the manufacturers do a lot of the heavy lifting for us, but it is important to understand what's going on for those occasions when things don't go to plan.

At the very least, we have added yet another tool to our growing set of ways to look at and understand power electronics systems.

As the number of models at our disposal grows, I am reminded of the saying, "all models are wrong, but some models are useful". Knowing just which model or tool to use in what circumstance is the mark of a true expert, and comes only with experience.

Next time, we will look at isolated DC-DC converters and reverse-engineer a commercial converter to see what we can learn from the professionals. PE

HAMMOND®
1551W IP68 miniature enclosures
Learn more: hammondmfg.com/1551w
uksales@hammondmfg.com
01256 812812

*Techno Talk continued from page 67...*

## Meet the XPUs

The first CPUs were, at their heart, scalar machines. Over time, to accelerate signal processing and graphics workloads, CPU designers added vector instructions and capabilities, enabling a single instruction to operate on multiple data elements in parallel. DSPs emerged to handle computationally intensive signal processing tasks that CPUs could not handle in real time.

DSPs feature specialised memory addressing modes and efficient vector-style processing for filters, transformations and modulation algorithms. They also have single-cycle multiply-accumulate (MAC) operations, which are key for digital filtering, convolution and other signal-processing algorithms.

As the use of graphical interfaces increased, it became increasingly untenable to use the main CPU for both general-purpose computation and graphical operations. The first GPUs implemented fixed-function graphics pipelines using arrays of lightweight CPU cores, each coupled with a small amount of local memory.

Over time, they evolved into programmable vector architectures optimised for data-parallel computation, of which graphics rendering was only one application. Today's GPUs extend this further with dedicated matrix and tensor engines that are used to accelerate AI workloads.

Neural Processing Units (NPUs) are primarily designed for matrix and tensor operations, although they may still include small scalar/vector units for control.

The term XPU is commonly used to mean 'some kind of processing unit', where "X" is a placeholder for the specific type. So XPU can be employed as an umbrella term for CPUs, DSPs, GPUs, NPUs and more.

## Chips and chiplets

Traditionally, XPUs were presented as individually packaged silicon dies ('silicon chips'). The next step was to implement multiple XPUs on the same piece of silicon, either as an array of the same XPU or a collection of different XPUs.

These days, it's becoming increasingly common to implement an XPU or a cluster of XPUs on a die called a chiplet, then connect multiple chiplets together and present them in a single package known as a multi-die system.

These days, there's a lot of talk about advanced packaging, which often involves throwing 2.5D, 3D, 3.5D and 4D qualifiers around.

2D refers to a conventional package containing a single die. Next up is 2.5D, which refers to multiple chiplets mounted on a common base layer called an interposer. This interposer may be organic (a circuit board) or semiconductor-based (silicon).

3D refers to multiple dies stacked on top of each other. These dies can be 10 to 20 times thinner than an untreated wafer die. In some cases, this stack ends up as the finished packaged device. For example, modern NAND flash memory uses dozens or even hundreds of vertically stacked memory layers fabricated in a single structure. The finished chip is packaged and used like any other component.

High-bandwidth memory (HBM) involves a 3D stack of DRAM dies with a logic die at the bottom, one or more HBMs can be mounted next to an XPU on an interposer, forming part of a multi-die package.

The term 3.5D is less standardised, but it refers to a combination of 3D stacking and side-by-side chiplet integration. There's also an informal 4D packaging term that involves adding dynamic behaviour over time, such as adaptive interconnects, reconfigurable chiplet fabrics, or runtime resource allocation.

## I've seen things…

The "Tears in Rain" monologue from the 1982 *Blade Runner* movie is commonly viewed as the defining moment in Rutger Hauer's acting career. It starts when he says, "I've seen things you people wouldn't believe…" I know how he felt, because I'm being introduced to unbelievable things on a daily basis.

For example, the folks at Hailo (**https://hailo.ai/**) recently introduced their Hailo-10H AI accelerator, whose core is an NPU-style architecture. Designed specifically to run neural-network inferencing (vision models, transformers, LLMs etc), the 10H delivers roughly 40 TOPS of INT4 performance using an architecture optimised for deep-learning workloads.

The mind-blowing thing to me is that they also offer the 10H as a Raspberry Pi HAT ('Hardware Attached on Top'). If my peers back in the 1970s would have freaked out over an Arduino Uno, they would have had a total meltdown if they'd been exposed to one of these RPi + AI HAT combos.

## Spatial dataflow fabric

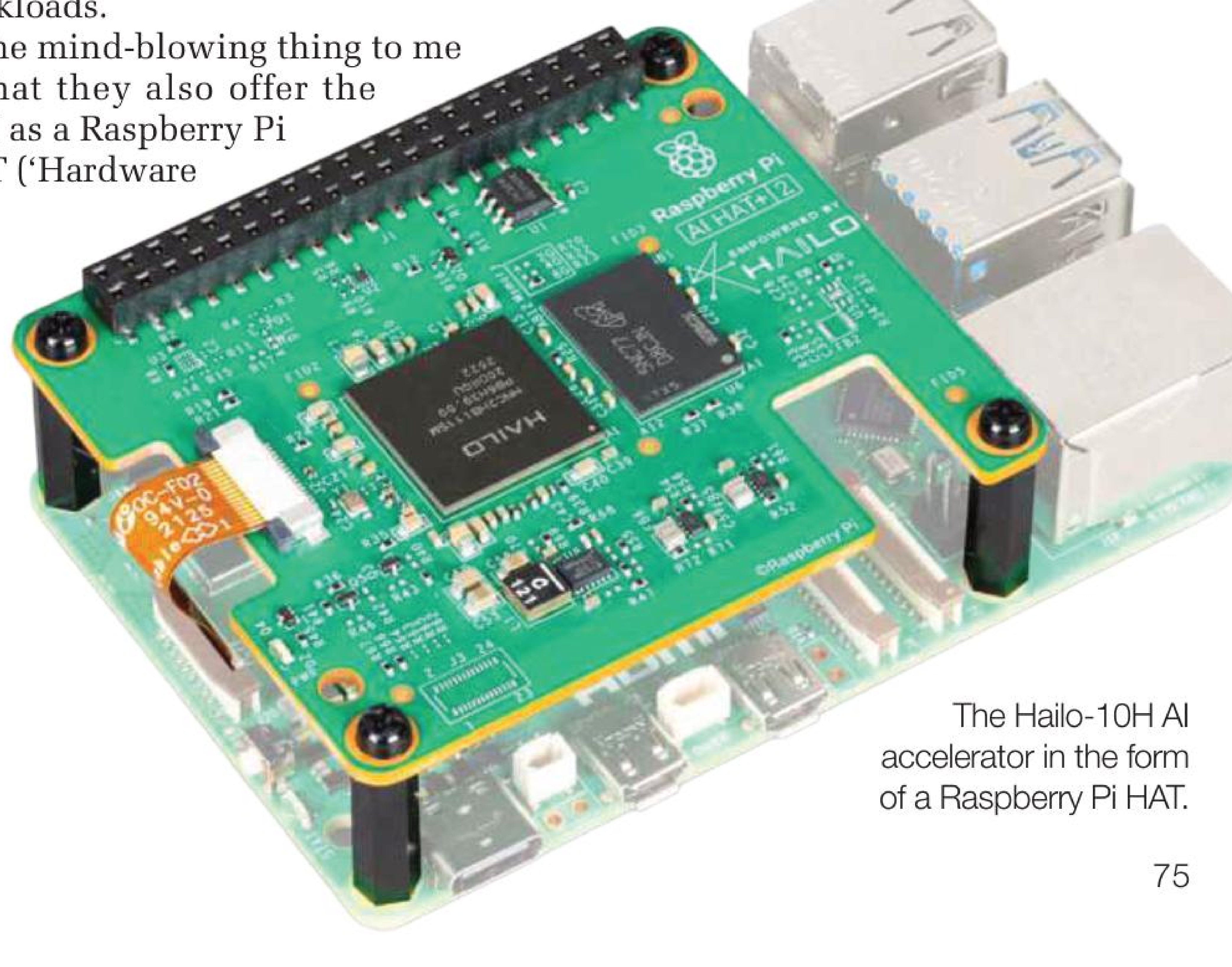
Something else I'm seeing is a lot of small companies coming up with new and innovative processor architectures to meet the needs of AI at the edge, where the internet 'rubber' meets the real-world 'road'. In many cases, these involve mounting a CPU and an NPU on the same piece of silicon, or as two chiplets in the same package. Then there are those who dare to be different.

The mission of Efficient Computer (**www.efficient.computer**) is to… well, the clue is in the name. Their Electron E1 processor is designed to run the kinds of mixed workloads common in cyber-physical systems—such as digital signal processing, control loops, sensor fusion, optimisation algorithms and AI inferencing—on a single programmable chip.

They have developed a radically different processor architecture based on a "spatial dataflow fabric". Instead of executing instructions sequentially like a conventional CPU, programs are compiled into a network of nodes across the chip, with data flowing directly between them.

By eliminating much of the energy wasted moving data between memory and compute units, the architecture can run mixed workloads (DSP, control, sensor fusion, analytics and AI) with dramatically higher efficiency, often delivering 10× to 100× better performance per watt than traditional embedded processors.

Whatever comes next will almost certainly make everything we've just talked about look quaint. And I, for one, can't wait to see it! PE



The Hailo-10H AI accelerator in the form of a Raspberry Pi HAT.

# Practical Electronics PCB SERVICE

| PROJECT | CODE | PRICE |
|---|---|---|
| **MAY 2026** | | |
| SSB Shortwave Receiver three PCB set | CSE250202-4 | £15 |
| Tool Safety Timer PCB | 10104251 | £3.99 |
| 433MHz Transmitter Module PCB | 15103251 | £2.99 |
| **APRIL 2026** | | |
| Versatile Battery Checker PCB | 11104251 | £3.99 |
| Versatile Battery Checker front panel (black, 0.8mm) | 11104252 | £4.99 |
| Power Rail Probe PCB | P9058-1-C | £3.99 |
| Pico 2 Audio Analyser PCB/panel (black) | 04107231 | £3.99 |
| **MARCH 2026** | | |
| USB-Programmable Frequency Divider PCB (blue) | 04108241 | £3.99 |
| Rotating Light PCB (black) | 09101251 | £2.99 |
| 433MHz Transmitter PCB | 15103251 | £2.99 |
| Arduino Uno Game Cartridge PCB (Cool Beans) | CBmar26uno | £3.99 |
| Arduino Nano Game Cartridge PCB (Cool Beans) | CBmar26nano | £2.99 |
| Silent Stomp Switch PCB pair (Audio Out) | AO-MAR26-* | £3.99 |
| **FEBRUARY 2026** | | |
| Power LCR Meter PCB | 04103251 | £7.99 |
| Versatile Waveform Generator PCB | 04104251 | £5.99 |
| **JANUARY 2026** | | |
| High-Bandwidth Differential Probe | 9015-D | £5.99 |
| NFC-Programmable IR Remote Keyfob | 15109231 | £2.99 |
| Pre-driver Overdrive Pedal main PCB | AO-JAN26-1 | £4.99 |
| Updated jack PCB for the Pre-driver | AO-JAN26-2 | £3.99 |
| **DECEMBER 2025** | | |
| Variable Speed Drive (black) | 11111241 | £11.99 |
| Digital Capacitance Meter | 04111241 | £5.99 |
| Battery-Powered Model Train transmitter | 09110241 | £3.99 |
| Battery-Powered Model Train through-hole receiver | 09110242 | £3.99 |
| Battery-Powered Model Train SMD receiver | 09110243 | £3.99 |
| Battery-Powered Model Train battery charger | 09110244 | £3.99 |
| Transmanium Fuzz Box jack PCB | AO-DEC-25 | £3.99 |
| **NOVEMBER 2025** | | |
| Capacitor Discharger | 9047-01 | £4.99 |
| 5MHz, 40A Current Probe | 9049-01 | £5.99 |
| Transmanium Fuzz Box main PCB | AO-NOV-25 | £4.99 |
| **OCTOBER 2025** | | |
| Seaside Sound Simulator PCB (blue) | 01111241 | £7.99 |
| Fender Jazz Bass pickguard | 23109241 | £5.99 |
| J&D Luthiers T-Style pickguard | 23109242 | £5.99 |
| Music Man Stingray pickguard | 23109243 | £5.99 |
| Fender Telecaster pickguard | 23109244 | £4.99 |
| 3D Printer Filament Dryer control PCB | 28110241 | £4.99 |
| **SEPTEMBER 2025** | | |
| Automatic LQ Meter PCB | CSE240203A | £5.99 |
| Automatic LQ Meter front panel (black) | CSE240204A | £5.99 |
| Compact Hi-Fi Headphone Amplifier PCB (blue) | 01103241 | £4.99 |
| Discrete Ideal Bridge Rectifier TH PCB | 18108241 | £2.99 |
| Discrete Ideal Bridge Rectifier SMD PCB | 18108242 | £2.99 |
| Dual Rail Supply Protector PCB | 18109241 | £3.99 |
| Audio Out Cascode Amplifier PCB | CASCODEA | £4.99 |
| Audio Out Cascode Amplifier Power Supply PCB | CASCODEP | £4.99 |
| **AUGUST 2025** | | |
| Micromite Explore-40 PCB | 07106241 | £2.99 |
| Micromite Explore-40 Audio Breakout PCB | 07101222 | £2.99 |
| 8-Channel Learning IR Remote Receiver PCB | 15108241 | £5.99 |
| FlexiDice PCB set | 08107241-2 | £4.99 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **JULY 2025** | | |
| Compact OLED Clock & Timer PCB | 19101231 | £4.99 |
| 180-230V DC Motor Speed Controller PCB | 11104241 | £9.99 |
| Mains Power-Up Sequencer PCB (same as Feb '25) | 10108231 | £11.99 |
| **JUNE 2025** | | |
| Styloclone (case version) | 23106241 | £6.99 |
| Styloclone (standalone version) | 23106242 | £7.99 |
| Dual Mini LED Dice (TH LEDs, white solder mask) | 08103241W | £2.99 |
| Dual Mini LED Dice (TH LEDs, black solder mask) | 08103241B | £2.99 |
| Dual Mini LED Dice (SMD LEDs, white solder mask) | 08103242W | £2.99 |
| Dual Mini LED Dice (SMD LEDs, black solder mask) | 08103242B | £2.99 |
| Dual Hybrid Supply regulator board | 18107211 | £5.99 |
| Dual Hybrid Supply front panel control board | 18107212 | £2.99 |
| Dual Hybrid Supply CPU board | 01106193 | £3.99 |
| Dual Hybrid Supply LCD adaptor | 01106196 | £2.99 |
| **MAY 2025** | | |
| Skill Tester 9000 | 08101241 | £9.99 |
| USB-C Serial Adaptor (black solder mask) | 24106241 | £2.99 |
| DC Supply Protector (SMD adjustable version) | 08106241 | £3.99 |
| DC Supply Protector (TH adjustable version) | 08106242 | £3.99 |
| DC Supply Protector (TH fixed version) | 08106243 | £3.99 |
| **APRIL 2025** | | |
| ESR Test Tweezers PCB set | 04105241+ | £5.99 |
| Fan Speed Controller Mk2 | 10104241 | £4.99 |
| Calibrated Microphone (SMD version) | 01108231 | £3.99 |
| Calibrated Microphone (TH version) | 01108232 | £3.99 |
| **MARCH 2025** | | |
| Pico Gamer | 08104241 | £9.99 |
| Compact Frequency Divider | 04112231 | £4.99 |
| **FEBRUARY 2025** | | |
| Laser Communicator transmitter | 16102241 | £3.99 |
| Laser Communicator receiver | 16102242 | £3.99 |
| Model Railway Points Controller transmitter | 09101241 | £3.99 |
| Model Railway Points Controller receiver | 09101242 | £2.99 |
| Mains Power-Up Sequencer | 10108231 | £11.99 |
| **JANUARY 2025** | | |
| USB to PS/2 Converter for VGA PicoMite | 07111231 | £3.99 |
| ps2x2pico USB to PS/2 Converter | 07111232 | £3.99 |
| Compact Microphone Preamplifier (case version) | 01110231 | £5.99 |
| Compact Microphone Preamplifier (integrated version) | 01110232 | £5.99 |
| **DECEMBER 2024** | | |
| Raspberry Pi Clock Radio PCB pair | 19101241/2 | £12.99 |
| VGA PicoMite (blue) | 07107221 | £5.99 |
| Secure Remote Switch receiver | 10109231 | £4.99 |
| Secure Remote Switch transmitter (module version) | 10109232 | £2.99 |
| Secure Remote Switch transmitter (discrete version) | 10109233 | £2.99 |
| **NOVEMBER 2024** | | |
| Ideal Diode Bridge Rectifiers (28mm square) | 18101241 | £3.99 |
| Ideal Diode Bridge Rectifiers (22mm square) | 18101242 | £3.99 |
| Ideal Diode Bridge Rectifiers (4-pin SIL) | 18101243 | £3.99 |
| Ideal Diode Bridge Rectifiers (tiny SOT-23) | 18101244 | £3.99 |
| Ideal Diode Bridge Rectifiers (standalone SMD) | 18101245 | £4.99 |
| Ideal Diode Bridge Rectifiers (standalone TH; 2oz copper) | 18101246 | £6.99 |
| Multi-channel Volume Control (volume PCB) | 01111221 | £4.99 |
| Multi-channel Volume Control (control PCB) | 01111222 | £4.99 |
| Multi-channel Volume Control (OLED PCB, black) | 01111223 | £4.99 |
| Coin Cell Emulator (black) | 18101231 | £3.99 |

PCBs for most recent *PE/EPE* constructional projects are available. From the July 2013 issue onwards, PCBs with eight-digit codes have silk screen overlays and, where applicable, are double-sided, have plated-through holes, and solder mask. They are similar to photos in the project articles. Earlier PCBs are likely to be more basic and may not include silk screen overlay, be single-sided, lack plated-through holes and solder mask.

Always check price and availability in the latest issue or online. A large number of older boards are listed for ordering on our website.

**In most cases we do not supply kits or components for our projects. For older projects it is important to check the availability of all components before purchasing PCBs.**

Magazine back issues are available – see https://www.electronpublishing.com/product-category/pe-back-issues-print-and-digital/

# Double-sided | plated-through holes | solder mask

| PROJECT | CODE | PRICE |
|---|---|---|
| **OCTOBER 2024** | | |
| TQFP-32 Programming Adaptor | 24108231 | £4.99 |
| TQFP-44 Programming Adaptor | 24108232 | £4.99 |
| TQFP-48 Programming Adaptor | 24108233 | £4.99 |
| TQFP-64 Programming Adaptor | 24108234 | £4.99 |
| 2m VHF FM Test Signal Generator | 06107231 | £4.99 |
| **SEPTEMBER 2024** | | |
| PIC Programming Adaptor | 24105231 | £4.99 |
| Salad Bowl Speaker crossover | 01109231 | £7.99 |
| **AUGUST 2024** | | |
| Arduino ESR Meter combined L/C-ESR | 04106182 | £9.99 |
| Arduino ESR Meter separate ESR | 04106183 | £7.99 |
| Watering System Controller | 15110231 | £11.99 |
| **JULY 2024** | | |
| Dynamic NFC Tag (22 × 31mm) | 06101231 | £5.95 |
| Dynamic NFC Tag (86 × 54mm) | 06101232 | £7.95 |
| Reciprocal Frequency Counter | CSE230101C | £8.95 |
| Reciprocal Frequency Counter front panel (black mask) | CSE230102 | £7.95 |
| Pi Pico-based Thermal Camera | 04105231 | £6.95 |
| **JUNE 2024** | | |
| Loudspeaker Test Jig | 04106231 | £12.95 |
| **MAY 2024** | | |
| GPS-Disciplined Oscillator | 04103231 | £12.95 |
| Dual RF Amplifier | CSE220602A | £5.95 |
| Songbird | 08103231 | £7.95 |
| **APRIL 2024** | | |
| Basic RF Signal Generator | CSE221001 | £8.99 |
| Basic RF Signal Generator front panel (black solder mask) | CSE220902B | £9.99 |
| Silicon Chirp Cricket | 08101231 | £7.99 |
| Test Bench 'Swiss Army Knife' | 04110221 | £9.99 |
| **MARCH 2024** | | |
| Digital Volume Control Potentiometer SMD version | 01101231 | £8.95 |
| Digital Volume Control Potentiometer through-hole version | 01101232 | £9.95 |
| Advanced SMD Test Tweezers body | 04106221 | £6.95 |
| Advanced SMD Test Tweezers legs (pair) | 04106212 | £7.95 |
| **FEBRUARY 2024** | | |
| Active Mains Soft Starter | 10110221 | £10.95 |
| **JANUARY 2024** | | |
| Q Meter | CSE220701 | £8.95 |
| Q Meter front panel (black solder mask) | CSE220704 | £7.95 |
| Raspberry Pi Pico W BackPack | 07101221 | £8.95 |
| **DECEMBER 2023** | | |
| Digital Boost Regulator | 24110224 | £8.95 |
| Dual-Channel Power Supply for Breadboards | 04112221 | £11.95 |
| Display Adaptor for the Breadboard PSU | 04112222 | £9.95 |
| **NOVEMBER 2023** | | |
| LC Meter Mk3 | CSE220503C | £9.95 |
| DC Supply Filter for vehicles | 08108221 | £8.95 |
| Discrete Audio Op Amp PCB | AO1-JUL23 | £9.95 |
| Discrete Audio Op Amp PCB + essential components | AO2-JUL23 | £17.95 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **OCTOBER 2023** | | |
| Buck/Boost Charger Adaptor | 14108221 | £9.95 |
| PIC Breakout Board for SOIC parts | 24110225 | £5.95 |
| PIC Breakout Board for DIP parts | 24110222 | £5.95 |
| AVR64DD32 Breakout board | 24110223 | £5.95 |
| Automatic Train Controller | 09109221 | £6.95 |
| Chuff Sound module | 09109222 | £6.95 |
| **SEPTEMBER 2023** | | |
| Mini LED Driver | 16106221 | £7.95 |
| New GPS-Synchronised Clock | 19109221 | £9.95 |
| **AUGUST 2023** | | |
| 110dB RF Attenuator | CSE211003 | £9.95 |
| Universal Battery Charge Controller (2023 update) | 14107192 | £9.95 |
| Wide-Range OhmMeter | 04109221 | £12.95 |
| **JULY 2023** | | |
| Multimeter Checker/Calibrator | 04107221 | £10.95 |
| **JUNE 2023** | | |
| Arduino Programmable Load | 04105221 | £8.95 |
| Buck-Boost LED Driver | 16103221 | £8.95 |
| **MAY 2023** | | |
| Precision AM-FM DDS Signal Generator | CSE211002 | £14.95 |
| Improved SMD Test Tweezers programmed PIC | 0410621PIC | £12.95 |
| AO 2x-dual-to-quad through-hole (pack of five) | AO1-MAY23 | £6.95 |
| AO 2x-dual-to-quad SMD (pack of five) | AO2-MAY23 | £6.95 |
| AO 2x-single-to-dual through-hole (pack of five) | AO3-MAY23 | £6.95 |
| AO 2x-single-to-dual SMD (pack of five) | AO4-MAY23 | £6.95 |
| **APRIL 2023** | | |
| 500W Amplifier Module | see p.22, April 2023 | |
| Clipping Indicator (per channel) | 01112211 | £7.95 |
| CD Welder Power Supply (one needed) | 29103221 | £9.95 |
| CD Welder Controller (one needed) | 29103222 | £9.95 |
| CD Welder Energy Storage module (several needed) | 29103223 | £7.95 |
| AO Universal Dual Op Amp Board | AO1-APR23 | £9.95 |
| AO Stereo RIAA precision passives kit for dual op amp | AO2-APR23 | £8.95 |
| **MARCH 2023** | | |
| Pico BackPack | 07101221 | £9.95 |
| Semaphore Signal (controller) | 09103221 | £7.95 |
| Semaphore Signal (blade) | 09103222 | £5.95 |
| **FEBRUARY 2023** | | |
| Tesla Coil driver board | 26102221 | £9.95 |
| Tesla Coil potentiometer board | 26102222 | £5.95 |
| Cooling Fan Controller & Loudspeaker Protector | 01102221 | £8.95 |
| Remote Gate Controller | 11009121 | £12.95 |
| **JANUARY 2023** | | |
| Classic LED Metronome – 8-LED | 23111211 | £7.95 |
| Classic LED Metronome – 10-LED | 23111212 | £8.95 |
| Multi-Channel Speaker Protector – 6 channel | 01101221 | £9.95 |
| Multi-Channel Speaker Protector – 4 channel | 01101222 | £7.95 |
| Remote Control Range Extender – IR-to-UHF | 15109212 | £5.95 |
| Remote Control Range Extender – UHF-to-IR | 15109211 | £7.95 |
| AO Universal Single Op Amp Board | AO1-JAN23 | £7.95 |
| **DECEMBER 2022** | | |
| Hummingbird Amplifier | 01111211 | £9.95 |
| SMD Trainer PCB | 29106211A | £8.95 |

All prices include VAT and UK p&p. Add £4 per *project* for post to Europe; £5 per *project* outside Europe.
Orders and payment should be sent to:

**Practical Electronics, Electron Publishing Ltd**
**113 Lynwood Drive, Merley, Wimborne, Dorset BH21 1UU**
**Tel 07918 614662    Email: pe@pemag.au**
**Online Shop: www.electronpublishing.com**

Sorry, cheques cannot be accepted as we are based in Australia and our UK bank account does not allow cheque deposits.

**NOTE:** Most boards are in stock and sent within seven days of order receipt. Please allow up to 28 days delivery if we need to restock.

# Practical Electronics PCB SERVICE

| PROJECT | CODE | PRICE |
|---|---|---|
| **NOVEMBER 2022** | | |
| USB Cable tester – main PCB | 04108211 | £12.95 |
| USB Cable tester – front panel | 04108212 | £5.95 |
| USB Cable tester – optional panel | SC5970 | £5.95 |
| Model Railway Carriage Lights – PCB | 09109211 | £6.95 |
| AO trnsfmr PCB – standard VTX-A range | VTX-101-007 | £6.95 |
| AO trnsfmr PCB – dual-outline VTX102-3001/101-3002 | VTX-Dual | £6.95 |
| **OCTOBER 2022** | | |
| SMD Test Tweezers – PCB and pair of tweezer arms | 04106211-2 | £11.95 |
| SMD Test Tweezers – programmed PIC12F1572-I/SN | 0410621A | £7.95 |
| Tele-com | 12110211 | £12.95 |
| **SEPTEMBER 2022** | | |
| Touchscreen Digital Preamp – main board | 01103191 | £12.95 |
| Touchscreen Digital Preamp – adaptor board pair | 01103192 | £5.95 |
| 20A DC Motor Speed Controller | 11006211 | £9.95 |
| **AUGUST 2022** | | |
| Multi-purpose Battery Manager – I/O Expander | 11104212 | £5.95 |
| Multi-purpose Battery Manager – Switch Module | 11104211 | £8.95 |
| Simple MIDI Music Keyboard (for 8 switches) | 23101213 | £6.95 |
| Nano Pong | 08105212 | £7.95 |
| **JULY 2022** | | |
| Silicon Labs AM/FM/SW Radio | CSE210301C | £10.95 |
| Level Crossing Controller | 09108211 | £6.95 |
| **JUNE 2022** | | |
| Full-wave Motor Speed Controller | 1010221 | £8.95 |
| PIC Programming Helper for 8-pin PICs only | 24106211 | £7.95 |
| PIC Programming Helper for 8, 14 or 20-pin PICs | 24106212 | £10.95 |
| Advanced GPS Computer | 05102211 | £9.95 |
| **MAY 2022** | | |
| Bus board PCB for Analogue Vocoder | AO1-MAY22 | £10.95 |
| Complete set of 14 PCBs for Analogue Vocoder | AO2-MAY22 | £97.95 |
| Programmed EEPROM for Digital FX Unit | FX1-MAY22 | £10.95 |
| Programmed PIC for Digital FX Unit w/ potentiometer | FX2-MAY22 | £8.95 |
| **APRIL 2022** | | |
| 64-key MIDI Matrix shield | 23101211 | £8.95 |
| 64-key MIDI Matrix switch board | 23101212 | £11.95 |
| High-current Battery Balancer | 14102211 | £10.95 |
| Digital FX Unit – using potentiometer | 01102211 | £9.95 |
| Digital FX Unit – using BCD switch | 01102212 | £9.95 |
| Universal Audio PSU | AO1-APR22 | £11.95 |
| **MARCH 2022** | | |
| Mini Isolated Serial Link | 24102211 | £5.95 |
| Busy Loo Indicator | 16112201 | £5.95 |
| Analogue Vocoder – Band-pass filter board | AO1-MAR22 | £9.95 |
| Analogue Vocoder – HP/LP filter board | AO2-MAR22 | £9.95 |
| **FEBRUARY 2022** | | |
| Arduino-based Power Supply | 18106201 | £9.95 |
| Battery Monitor Logger | 11106201 | £10.95 |
| Electronic Wind Chimes | 23011201 | £10.95 |
| Analogue Vocoder – Driver Amplifier | AO-FEB22 | £8.95 |
| **JANUARY 2022** | | |
| Vintage battery Radio Li-ion Power Supply | 11111201 | £9.95 |
| MiniHeart: A Miniature Heartbeat Simulator | 01109201 | £8.95 |
| **DECEMBER 2021** | | |
| AM/FM/SW Digital Receiver | CSE200902A | £13.95 |
| Balanced Input and Attenuator for USB CODEC | 01106202 | £11.95 |

| PROJECT | CODE | PRICE |
|---|---|---|
| **NOVEMBER 2021** | | |
| Dual Battery Lifesaver | 11111202 | £6.95 |
| **OCTOBER 2021** | | |
| Mini Wi-Fi LCD BackPack | 24106201 | £8.95 |
| **SEPTEMBER 2021** | | |
| USB SuperCodec PCB | 01106201 | £14.95 |
| Audio DDS Oscillator PCB | 01110201 | £5.95 |
| Audio DDS Oscillator rotary encoder | 01110201-ENC | 6.95 |
| Programming Adaptor Board for Audio DDS Oscillator | 01110202 | £5.95 |
| High-power Ultrasonic Cleaner main PCB | 04105201 | £14.95 |
| High-power Ultrasonic Cleaner front-panel PCB | 04105202 | |
| Night Keeper Lighthouse PCB | 08110201 | £5.95 |
| **AUGUST 2021** | | |
| Ol' Timer PCB | 19104201 | £11.95 |
| Ol' Timer 8x8 RGB LED module using WS2812B | 19104201-88 | £8.95 |
| Ol' Timer set of acrylic case pieces and spacer | 19104201-ACR | £8.75 |
| Ol' Timer DS3231 RTC IC wide SOIC-16 | 19104201-RTC | £5.95 |
| Wideband Digital RF Power Meter | 04106201 | £9.75 |
| Switchmode 78xx regulators ***(PACK of 5!)*** | 18105201 | £7.95 |
| Cool Beans SMAD display | CB-AUG21 | £11.95 |
| **JULY 2021** | | |
| ATtiny816 Breakout / Dev Board w/ Capacitive Touch | 24110181 | £9.75 |

**For the many earlier PCBs that we stock please see the PE website: www.electronpublishing.com**

## *PE/EPE* PCB SERVICE

Order Code Project Quantity Price

Name

Address

Tel

Email

Card No

Valid From Expiry Date

Card Security No

**You can also order PCBs by phone, email or via the shop on our website: www.electronpublishing.com**

**No need to cut your issue – a copy of this form is just as good!**

Silicon Chip kits & parts for projects in this issue (posted from Australia):

| | | |
|---|---|---|
| 433MHz Transmitter Module kit | https://siliconchip.au/Shop/20/7447 | AUD $24 (about £12.50) + P&P |
| Tool Safety Timer programmed PIC | https://siliconchip.au/Shop/9/7478 | AUD $12 (about £6.25) + P&P |
| SI5351A clock generator module for SSB Shortwave Receiver | https://siliconchip.au/Shop/7/2737 | AUD $9 (about £5.00) + P&P |
| T50-6 toroidal core for SSB Shortwave Receiver | https://siliconchip.au/Shop/7/7504 | AUD $6 (about £3.00) + P&P |
| BF998 dual-gate Mosfet (SOT-143) for SSB Shortwave Receiver | https://siliconchip.au/Shop/7/2750 | AUD $3 (about £1.50) + P&P |
| Backlit 16×2 alphanumeric LCD module for SSB Shortwave Receiver | https://siliconchip.au/Shop/7/5759 | AUD $12 (about £6.25) + P&P |

20% VAT is included in the above price (it will be removed for customers outside the UK). Postage & packing cost is typically around £12-15.
Please allow 1-2 weeks for delivery (longer in some cases).

# CLASSIFIED ADVERTISING

# Practical Electronics

If you want your advertisements to be seen by the largest readership at the most economical price then our classified page offers excellent value.

The rate for semi-display space is £10 per centimetre high, with a minimum height of 2·5cm. All semi-display advertisements have a width of 5.5cm. The prepaid rate for classified adverts is 40p per word (minimum 12 words).

For rates, further information on display and classified advertising or to book an ad, please email us at pe@pemag.au

**Practical Electronics reaches more UK readers than any other UK monthly hobby electronics magazine. Our sales figures prove it.**

**We have been the leading monthly magazine in this market for more than thirty years.**

**W5SWL Electronics**

Technical Products and Small Component Parts, RF and Audio Connectors, Hobby Electronics, Ham Radio to Microwave Products, General Hobby Supplies, Technical Books, Tools, and More. **Located in the United States.**

Visit **www.W5SWL.com**

**BOWOOD ELECTRONICS LTD**

*Suppliers of Electronic Components*

**www.bowood-electronics.co.uk**

Unit 10, Boythorpe Business Park, Dock Walk, Chesterfield,
Derbyshire S40 2QR. Sales: 01246 200 222

Send large letter stamp for Catalogue

**COAST ELECTRONICS**

BREAKOUTS-COMPONENTS-
CONTRACT DESIGN-3D PRINTER PARTS-
MUSICAL-MICROCONTROLLERS
**WWW.COASTELECTRONICS.CO.UK**

## MISCELLANEOUS

**VALVES AND ALLIED COMPONENTS?** For free stock list and/or advice, please contact me: geoffdavies337@gmail.com Telephone: 01788 574774

**PIC DEVELOPMENT KITS, DTMF kits** and modules, CTCSS Encoder and Decoder/Display kits.
Visit www.cstech.co.uk

**Electrical Industries Charity (EIC)**
We help people working in the electrical, electronics and energy community as well as their family members and retirees.

We use workplace programmes that give the industry access to financial grants and a comprehensive range of free and confidential services.
**www.electricalcharity.org**

electrical industries charity

# ADVERTISING INDEX

| | |
|---|---|
| AOSHOP | 53 |
| ENIGMA COMPONENT SHOP | 5 |
| HAMMOND MANUFACTURING | 74 |
| MICROCHIP TECHNOLOGY | Outer back cover |
| PCBWAY | Inner front cover |
| PDF BACK ISSUES | 12, 59 |
| PO LABS | 37 |
| TEACH-IN BUNDLES | 45 |

**Advertising mailing address**

Electron Publishing Australia
PO Box 194
Matraville NSW 2036
Australia

**Email** pe@pemag.au

**Web** www.electronpublishing.com

For editorial contact details, see page 2.

# Next Month – in the June issue

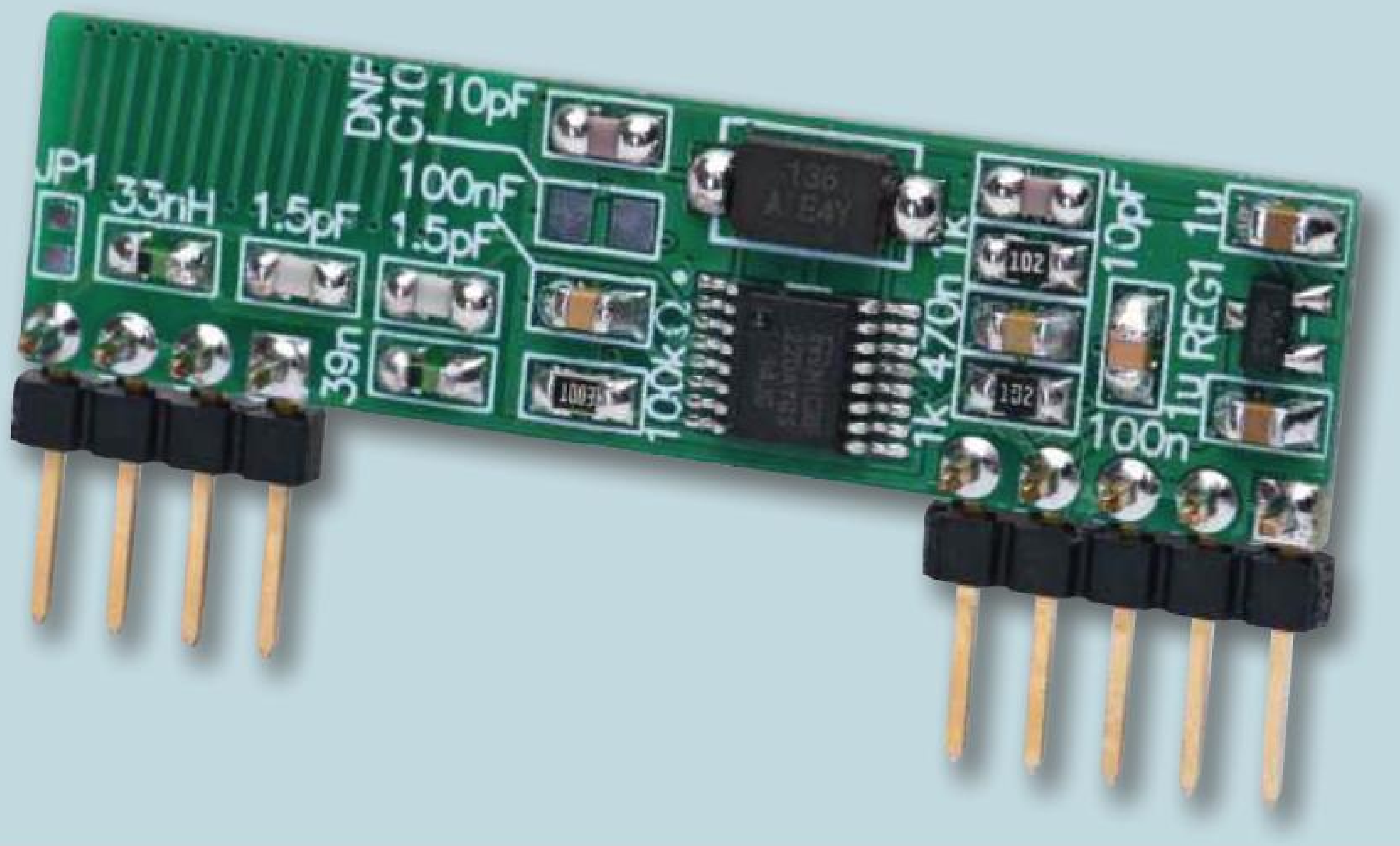
12
1
2
3
4
5
6
7
8
9
10
11
SILICON CHIP
AM
PM
A
B

**USB Power Adaptor**

A simple and inexpensive way to add a USB socket to a 5V-powered board

**RGB LED 'Analog' Clock**

Coloured LEDs light up around the edge to tell the time with a spinning chaser pattern.



**DIY LPD433 Receiver Module**

It works well with the 433MHz transmitter in this issue. It has great performance with an onboard trace antenna but also supports external antennas.

**On sale 5th of May, 2026**

Content may be subject to change

## Practical Electronics

## Subscriptions hotline

To subscribe or renew online, paying by credit/debit card or PayPal, please visit our shop at: www.electronpublishing.com

For those wishing to pay by direct debit, cheque or over the telephone, please call 'Select Publisher Services' on 01202 087631, or send a cheque (payable to: '**Select Publisher Services Ltd**') with your details to:

Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH, United Kingdom

Remember, we print the date of the last issue of your current subscription in a box on the address sheet that comes with your copy.

Digital subscribers, please call 07918 614662 or visit: www.electronpublishing.com

Published on approximately the first Thursday of each month by Silicon Chip Publications, PO Box 194 Matraville NSW 2036 Australia. Printed in England by The Manson Group, Reynolds House, 8 Porters Wood, Valley Road Industrial Estate, St Albans AL3 6PZ. Distributed by Seymour, 86 Newman St, London W1T 3EX. Subscriptions UK: £33.99 (6mo); £59.99 (12mo.); £114.99 (2yrs). EUROPE: airmail service, £38.99 (6mo); £69.99 (12mo); £129.99 (2yrs). REST OF THE WORLD: airmail service, £46.99 (6mo); £84.99 (12mo); £164.99 (2yrs). Make payments to 'Practical Electronics', Practical Electronics Subscriptions, PO Box 6337, Bournemouth BH1 9EH United Kingdom. Email: pesubs@selectps.com. PRACTICAL ELECTRONICS is sold subject to the conditions that it shall not, without the written consent of the Publishers first having been given, be lent, resold, hired out or otherwise disposed of by way of Trade at more than the recommended selling price on the cover, and that it shall not be lent, resold, hired out or otherwise disposed of in a mutilated condition or in any unauthorised cover by way of Trade or affixed to or as part of any publication or advertising, literary or pictorial matter.

# Did you know our online shop now sells the current issue of *PE* for £5.99 inc. p&p?

Practical Electronics

The UK's premier electronics, computing and maker magazine

**Max's Cool Beans** The classic Snake game & a VFD timer

**Teach-In 12** Part 7: spectrum and vector network analysis

**Audio Out** All-pass filters for speaker alignment

**Circuit Surgery** Better ways to stabilise op amps

Build a Single Sideband Shortwave Receiver with digital tuning

DIY LPD433 transmitter module

A motion-sensing Safety Timer for tools like soldering irons

**PLUS!**

Power Electronics, part two – Understanding DC/DC converters

Techno Talk – Tensor processors and Raspberry Pi AI HATs

The Fox Report – Freely's success will depend on good hardware

May 2026 £5.99

www.electronpublishing.com | @practicalelec | practicalelectronics

Practical Electronics

**Audio Out** All-pass filters: unsung heroes

**Circuit Surgery** Stabilising op amps using output resistors

Apr 2026 £5.99

You read that right! We now sell the current issue of your favourite electronics magazine for exactly the same price as in the High Street, but we deliver it straight to your door – and for UK addresses we pay the postage. No need to journey into town to queue outside the newsagent. Just go to our website, set up an account in 30 seconds, order your magazine and we'll do the rest.

# www.electronpublishing.com

ENERGY
STORAGE SYSTEM

# Engineered for Power Efficiency

## See Every Watt, Save Every Microamp

Tired of burning significant power just to measure power? Want real-time alerts for system protection without sacrificing efficiency? With the PAC1711 and PAC1811, you no longer have to sacrifice battery life just to track your system’s power usage.

The PAC1711 and PAC1811 reduce power consumption by 50% compared to typical 12-bit and 16-bit digital power monitors. Both devices let you monitor power status without draining your battery, maximizing uptime and efficiency in every battery-dependent or backup power application.

### Key Features

- Ultra-Low Power Consumption: The PAC1711 operates at just 100µA, only 50% of the active current of similar 12-bit monitors, while the PAC1811 operates at just 140µA, only 50% of the active current of similar 16-bit monitors.
- Real-Time System Alerts: Up to 2 alert outputs for immediate notification of out-of-limit power events.
- Effortless Integration: PAC1711 and PAC1811 are pin- and footprint-compatible with SOT23-8 for easy upgrades and migration.
- Precision Monitoring without resolution loss at higher sample rates on both the PAC1711 at 12-bits and the PAC1811 at 16-bits.
- Enhanced System Protection: Immediate alerts help safeguard your devices.

MICROCHIP
PAC1711T
MICROCHIP
PAC1811

Empower your next project with the PAC1711 and PAC1811, where efficiency, safety, and ease of integration come together.

MICROCHIP

microchip.com/MorePowerInsight

The Microchip name and logo and the Microchip logo are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. All other trademarks are the property of their registered owners.
© 2026 Microchip Technology Inc. All rights reserved.
MEC2636A-UK-01-26